Sous-typage coercitif en présence de réductions non-standards dans un système aux types dépendants

Lionel Marie-Magdeleine

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Discipline ou spécialité :

Informatique

Présentée et soutenue par :
Lionel Marie-Magdeleine

le : vendredi 11 décembre 2009

Titre :

Sous-typage coercitif en présence de réductions non-standards dans un système aux types dépendants

---

Ecole doctorale :
Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :
IRIT-UPS

Directeur(s) de Thèse :

Sergei Soloviev

Rapporteurs :
Giuseppe Longo, Directeur de Recherche, Ecole Normale Supérieur
Alexandre Dikovsky, Professeur, Université de Nantes

Autre(s) membre(s) du jury

Jean-Paul Bodeveix, Professeur, Université Paul Sabatier, Président
Anne Preller, Professeur émérite, Université de Montpellier
Sergei Soloviev, Professeur, Université Paul Sabatier

Sous-typage coercitif en présence de réductions non-standards dans un système aux types dépendants

Lionel Marie-Magdeleine

Remerciements

Je tiens tout d'abord à exprimer ma reconnaissance et ma gratitude à Sergei Soloviev, mon directeur de recherche, qui a su me faire confiance et me guider tout au long de mes recherches. Je mets également en avant sa gentillesse et sa disponibilité.

Je remercie Giuseppe Longo et Alexandre Dikovsky d'avoir accepter la charge de rapporteur, ainsi qu'Anne Preller et Jean-Paul Bodeveix de m'avoir fait l'honneur de participer à mon jury de thèse. Je les remercie tous très vivement d'avoir accepté de se pencher sur mes travaux.
Je voudrais également dire merci à Louis Féraud de m'avoir toujours ouvert sa porte pour un conseil ou pour toute aide ou renseignement quelqu'il soit.

Je suis très reconnaissant envers Ralph Matthes et Zhaohui Luo pour le temps qu'ils m'ont consacré, l'intérêt qu'ils ont porté à mes travaux et leurs conseils scientifiques qui m'ont beaucoup éclairé. Je remercie également Healfdene Goguen dont l'étude de sa thèse représente une partie de mes travaux.

Je remercie Claire-Emmanuelle pour son soutien et ses nombreuses relectures, Arnault pour son soutien et ses conseils avisés.

Je remercie ma famille, mon père, ma mère et mon frère pour leur soutien inconditionnel et leurs nombreux encouragements dans lesquels j'ai puisé toute l'énergie dont j'avais besoin.

Je remercie le conseil régional de la Martinique pour m'avoir financé ma dernière année de thèse.

Bien-sûr, je n'oublie pas ce qui a contribué à rendre mon séjour en tant que thésard agréable ; je pense à l'équipe ACADIE, au rituel du thé dans le bureau de Louis, à tous mes camarades thésards, notamment Antoine mon compagnon de bureau que je remercie pour notre bonne entente.

Pour finir, j'ai une pensée pour Claire-Emmanuelle et mon frère Loïc qui sont conscients de ce que représente le fait de faire un thèse.
Foss epi couraj.

Simin maché pou ayin ke domi pou ayin

Ce qui est en Haut est comme ce qui est en Bas ; ce qui est en Bas est comme ce qui est en Haut.

Hermès Trimégiste

Table des matières

Introduction

La théorie des types a été proposée comme fondement des mathématiques constructives et comme cadre pour construire des logiciels certifiés et pouvoir extraire des programmes à partir de preuves. De nos jours la plupart des théories de types qui fait l'objet d'études approfondies intègre ou pas une forme d'imprédicativité, des types inductifs, des types coïnductifs, des types dépendants, la notion de sous-typage et des relations de réductions élaborées. On distingue particulièrement le calcul des constructions inductives de Coquand et Huet [CH88], et la "Unified Type Theory" (définie au sein du "Logical Framework") de Luo [Luo94b].

Ces systèmes ont fait l'objet de nombreuses implémentations sous la forme de langages fonctionnels expressifs et intégrés dans des logiciels d'assistance à la preuve. On peut particulièrement citer Coq, Plastic, Lego. Un assistant de preuve est un logiciel permettant l'écriture et la vérification de preuves mathématiques, soit sur des théorèmes au sens usuel des mathématiques, soit sur des assertions relatives à l'exécution de programmes informatiques. Dans le cas de la conception des programmes informatiques, plutôt que de coder un programme puis de vérifier à posteriori qu'il est conforme à sa spécification, les assistants à la démonstration permettent de prouver préalablement le bien fondé de la spécification, en fournissant une preuve de cette spécification qui n'est autre que le programme lui même.

De manière générale, la théorie des types constitue une discipline étudiant des formalismes contenant à la fois une dimension calculatoire et une dimension logique liées par l'isomorphisme de Curry-Howard. Elle se situe au croisement de la logique, des mathématiques et de l'informatique et peut, dans ce dernier cadre, servir de support au développement de programme "zéro faute".

L'approche du sous-typage coercitif comme mécanisme d'abréviation a été proposée et développée par Z.Luo [Luo97c]. Par exemple si f est une fonction de type $A \to B$, A_0 un sous-type de A, t un élément de type A_0 et c de type $A_0 \to A$ une coercion, l'expression $f(t)$ est vue comme une abréviation de

$f(c(t))$. Ce fait est exprimé par une règle de définition coercitive qui permet d'obtenir $f(t) = f(c(t))$. Une coercion est une application particulière qui définit comment les éléments de A_0 correspondent aux éléments de A, souvent mais pas nécessairement c peut être vu comme un plongement. Cette approche du sous-typage dans le cas des systèmes aux types dépendants est intéressante car le sous-typage peut grandement améliorer l'efficacité de nombreux systèmes d'assistance à la preuve basés sur les systèmes aux types dépendants. En revanche elle est plus compliquée à mettre en œuvre .

Une des propriétés principales du sous-typage coercitif comme mécanisme d'abréviation est la conservativité de la théorie de types étendue par sous-typage coercitif par rapport à la théorie de base. Le retour de la théorie étendue à la théorie de base est assuré par l'algorithme d'insertion des coercions. Le résultat est un jugement dans la théorie de base. La conservativité implique qu'il n'est pas possible par ce mécanisme d'obtenir des jugements qui ne sont pas dérivables dans la théorie de base. Dans [SL98a], il a été montré l'importance de la cohérence des coercions pour la preuve de conservativité. La notion de cohérence signifie, premièrement qu'il est impossible d'avoir deux coercions différentes $c_1, c_2 : A_0 \to A$ pour le même sous-type A_0 de A, deuxièmement qu'il n'existe pas de coercion d'un type A vers lui même, autrement dit le type A ne peut pas être considéré comme un sous-type de A lui-même [1]. Considérons l'exemple suivant : le type des Anneaux $(A, +, *)$ est un sous-type des Groupes par rapport à l'opérateur "+" et des Monoïdes par rapport à l'opérateur "*". En même temps le type des Groupes est un sous-type des Monoïdes par rapport à l'opérateur "+". On peut définir les coercions suivantes :

$$\begin{tikzcd} \text{Anneaux} \arrow[r, "c_1"] \arrow[dr, "c_3"'] & \text{Groupe} \arrow[d, "c_2"] \\ & \text{Monoïde} \end{tikzcd}$$

Évidemment on a $c_2 \circ c_1 \neq c_3$.
Les définitions des types de divers systèmes algébriques basées sur l'utilisation de Σ-types dans un système aux types dépendants peuvent être trouvées dans [Luo94b]. Considérons un autre exemple montrant la relation entre des types encore plus communs : les codes ASCII, les entiers (Nat) et les chaînes de caractères (String). On peut donc définir les coercions suivantes :

[1]Une autre possibilité est de prendre la fonction identité sur A, c'est-à-dire $\lambda x^A.x : A \to A$ comme terme de coercion, cela représente un changement technique mineure.

$$\begin{tikzcd} \text{ASCII} \arrow[r, "c_1"] \arrow[dr, "c_3"'] & \text{Nat} \arrow[d, "c_2"] \\ & \text{String} \end{tikzcd}$$

Ces coercions peuvent être définies de sorte que si on considère le code ascii 65 noté Ascii(65), l'entier 65 noté Nat(65), les chaînes de caractères "A" et "65" notées respectivement String(A) et String(65), on ait :

$$\begin{tikzcd} \text{ASCII(65)} \arrow[r, "c_1"] \arrow[dr, "c_3"'] & \text{Nat(65)} \arrow[d, "c_2"] \\ & \text{String(65)} \end{tikzcd}$$

Évidemment dans ce cas on a également $c_2 \circ c_1 \neq c_3$.
Les deux exemples précédents nous exposent des cas où la propriété de conservativité n'est pas vérifiée car la condition de cohérence ne l'est également pas.
La non-conservativité due à l'absence de cohérence peut être une conséquence de l'utilisation de la règle de définition coercitive en combinaison avec les autres règles du système. Si on a deux coercions $c_1, c_2 : A_0 \longrightarrow A$, où $c_1 \neq c_2$ dans la théorie de base, on peut alors déduire $f(c_1(t)) = f(t)$ et $f(t) = f(c_2(t))$. Si on prend $f \equiv \lambda x^A.x$ et une variable $y : A_0$ avec $y \notin FV(c_1, c_2))$, on peut facilement déduire $c_1 = c_2$, ce qui mène à une contradiction. Les deux exemples plus haut montrent aussi une des sources possibles de non-cohérence : tandis que les coercions de base (définies par exemple par l'utilisateur) peuvent être uniques par rapport à chaque paire de sous-type A_0 et de type A ($A_0 < A$), les compositions de coercions quant à elles ne sont pas forcément uniques. Il existe des règles, comme celles concernant les coercions entre types-produits dans la théorie aux types dépendants, permettant de créer de nouvelles coercions . L'application de toutes ces règles pourrait en principe produire des systèmes de coercions non cohérents. Une partie importante d'une preuve de conservativité doit assurer la cohérence de l'ensemble des coercions, pas seulement des coercions de base.

Dans [SL00], on retrouve la preuve de conservativité pour le "Logical Framework" LF proposé par Z.Luo. LF est un cadre logique permettant de déclarer des théories de types par ajout de constantes et de règles de calcul. L'idéal serait de pouvoir généraliser la preuve de conservativité à LF et à toutes les théories pouvant être déclarées dans LF. Jusqu'à maintenant, il existait une preuve incomplète de conservativité pour le système UTT esquissée dans [SL00]. UTT est une théorie déclarée dans LF qui comprend entre autre, des types dépendants, des types inductifs, un univers prédicatif, un univers imprédicatif avec des constantes pour une logique interne du second ordre. Prouver le théorème de conservativité pour UTT est donc un pas vers la généralisation de ce théorème aux théories pouvant être déclarées dans LF. C'est justement une des tâches à laquelle s'adonne cette thèse.

D'autre part, en réalité la classe des systèmes de coercions cohérents est relativement faible pour les théories de types n'ayant que des réductions standards comme UTT. Augmenter le nombre d'égalités intensionnelles de ces théories de type permettrait d'enrichir la classe des systèmes de coercions cohérents. Pour se faire, une possibilité serait d'intégrer sous forme de réductions certaines égalités extensionnelles. C'est un problème sur lequel nous allons également nous pencher.

Un des problèmes qui est levé quand une théorie des types est utilisée pour décrire des théories qui combinent de la logique et du calcul, est le problème de la relation entre l'égalité intensionnelle et extensionnelle. Ce problème est familier à tous les utilisateurs d'assistant à la preuve basée sur la théorie des types (comme Lego, Coq ou Isabelle). Dans les théories de types, l'égalité intensionnelle est définie par la conversion des termes basée sur les réductions tandis que l'égalité extensionnelle est prouvée à l'aide de mécanismes internes et externes. Pour prouver une formule comme $\lambda x.(f(x) = g(x))$, la plupart des assistants à la preuve basée sur la théorie des types possède des logiques internes et la preuve est représentée par un terme. L'existence d'un tel terme n'implique bien-sûr pas que les deux fonctions sont convertibles par des réductions, c'est-à-dire intensionnellement égales. Cette différence est manifeste même dans les cas simples. Considérons un système de reductions standards dans le lambda calcul simplement typé avec types inductifs. Ce système comprend la β, η-réduction ainsi la ι-réduction pour les opérateurs de récursion. Si on considère le type $Nat = ind(\alpha).(0 : \alpha, succ : \alpha \to \alpha)$[2], la fonction d'identité $\lambda x : Nat.x : Nat \to Nat$ est extensionnellement égale à $\lambda x : Nat.(0 + x) : Nat \to Nat$, (où $+$ est définie par une récursion sur le second opérande), mais elle ne l'est pas par application de réductions. En effet la ι-réduction n'est applicable que lorsque l'argument est 0 ou de la forme $succ(t)$, mais pas lorsque l'argument est une variable. Bien sûr, il existe un terme de preuve pour $\lambda x : Nat.x = \lambda x : Nat.(0 + x)$, mais ce terme devra être réutilisé si on doit produire une preuve d'égalité (probablement plus longue) qui dépend de l'égalité dont le terme est la preuve. Ce problème surgit dans toutes les situations faisant intervenir égalité extensionnelle et intensionnelle et représente

[2] Par la suite nous utiliserons la notation de LF

un sérieux inconvénient pour les assistants à la preuve basés sur la théorie des types. L'importance de ce problème est soulignée par le fait que vouloir ajouter toutes les égalités extensionnelles dans le système de reduction crée un mauvais système de réductions car les propriétés comme l'unicité de la forme normale et la normalisation forte ne sont pas garanties dans le cas des systèmes avec des types dépendants (comme le Calcul des Constructions utilisé dans COQ ou UTT utilisé dans Lego). Même le typage peut devenir indécidable.

L'approche que nous considérons dans cette thèse est basée sur l'extension de systèmes de réductions, par des réductions représentant des égalités extensionnelles et préservant les bonnes propriétés du système de reduction. La normalisation forte et la confluence (Church-Rosser) devront être prouvées une seule fois. Ensuite l'utilisation du lambda calcul suivra le même schéma : certaines égalités seront prouvées par réduction (ce qui est plus efficace), et les autres nécessiteront de trouver une preuve sous forme de terme. Dans ce dernier cas, la classe des égalités intensionnelles et extensionnelles étant changée, il y a plus d'égalités intensionnelles, le système gagne en efficacité.

La principale nouveauté dans cette thèse est que l'on considère une extension des réductions standards dans un système aux types dépendants qui se nomme UTT (Unified Type Theory) et qui est utilisé dans des assistants à la preuve comme Lego et Plastic. La réduction que nous rajoutons concerne une égalité extensionnelle pour les fonctions entre types finis et devrait fournir un mécanisme efficace pour le traitement des égalités pour un fragment de UTT.

Avec l'ajout d'une nouvelle réduction, les propriétés de normalisation forte et de confluence ne sont plus garanties. Or il est impératif de pouvoir garantir ces propriétés afin conserver un système fonctionnel. Healfdene Goguen dans sa thèse [Gog94] sous la direction Rod Burstall et Zhaohui Luo, prouve que le calcul UTT possède certaines propriétés dont la normalisation forte, la préservation du type et la confluence. Pour se faire, il utilise une méthode basée sur l'emploi d'une sémantique opérationnelle typée. Nous rappelons que le système étendu que nous considérons dans cette thèse, n'est autre que le système UTT auquel nous rajoutons une règle d'égalité. C'est tout naturellement que nous utiliserons la méthode employée par Goguen en l'adaptant afin de prouver que le système étendu vérifie les propriétés de normalisation forte, préservation du type et de confluence.

Nous proposons dans cette thèse d'enrichir d'une nouvelle réduction pour les fonctions entre les types finis, la ϑ-réduction, un système aux types dépendants UTT. Ce nouveau système se nomme UTT^{τ}. Nous faisons remarquer que l'ajout la ϑ-réduction se traduit par l'ajout d'une règle d'inférence d'égalité, la règle (ϑ-Eq), au système UTT. Une fois la ϑ-réduction ajoutée, nous commençons par prouver que le nouveau système UTT^r vérifie bien les propriétés de normalisation forte, de préservation du type et de confluence. Ensuite nous effectuons la preuve complète de conservativité pour le calcul UTT^r enrichi du sous-typage coercitif. Cette même preuve nous permettra de déduire que la conservativité est également vérifiée pour le calcul UTT enrichi du sous-typage coercitif. Il faut tout de même noter que la preuve de conservativité pour des systèmes étendus par le sous-typage coercitif ne dépend pas des propriétés de normalisation forte, de préservation du type et de confluence de ces systèmes complétés par de nouvelles réductions.

L'intérêt du calcul UTT^r est que pour un fragment de UTT concernant les fonctions entre types finis, il améliorera grandement l'efficacité des assistants à la preuve basés sur les systèmes aux types dépendants. Cette efficacité sera améliorée d'une part par l'ajout de cette égalité sous forme d'égalité intensionnelle, d'autre part par l'augmentation des systèmes de coercions cohérents pour le sous-typage. Étant donné que Les types finis représentent des ensembles finis, le calcul UTT^r permettra de traiter plus aisément les problèmes d'analyse combinatoire, de manipulation de graphes etc.

Dans le chapitre 1, nous commençons par introduire les notions d'interprétation, de modèle et de sous-typage qui représentent le contexte de cette thèse. Dans le chapitre 2, nous présentons et fournissons toutes les définitions permettant d'appréhender le système UTT^r, qui rappelons-le, est une extension du système UTT par une égalité sur les fonctions entre types finis. Dans ce même chapitre, nous énonçons et démontrons les propriétés basiques du système que l'on considère. Puis nous réalisons la preuve que le système UTT^r possède les propriétés de normalisation forte, de préservation du type et de confluence, en effectuant les modifications nécessaires de la preuve de Goguen. Dans le chapitre 4, nous présentons le résultat le plus important qui est l'étude du système UTT^r enrichi du sous-typage coercitif. Nous introduisons ce calcul en présentant le sous-typage coercitif que nous considérons, à savoir celui présenté dans [Luo97c]. Ensuite, nous réalisons la preuve de conservativité pour deux théories déclarées dans le "Logical Framework" LF enrichi du sous-typage coercitif, à savoir UTT et UTT^r.

Chapitre 1

Contexte

1.1 Les modèles

Afin d'étudier le comportement et les propriétés des théories de types, il est souvent nécessaire de considérer leurs sémantiques. Les techniques syntaxiques peuvent être difficiles à mettre en œuvre, et certaines propriétés ne peuvent pas être démontrées uniquement en tenant compte de la syntaxe des termes. La notion de modèle a été introduite justement pour l'étude sémantique des théories de types. La plupart des preuves de normalisation forte passe par l'utilisation d'un modèle sémantique. Nous effectuons d'abord un bref rappel des principaux modèles du Calcul des Constructions avec univers $CC\omega$[1][Luo94b] qui permettra de mieux appréhender l'étude des propriétés de UTT^r du chapitre 2. Nous limiterons notre présentation à deux modèles standards pour $CC\omega$ à savoir : le modèle ensembliste [Acz98, Wer97] et le modèle catégorique [Luo94b]. La présentation de ces deux modèles s'inspire de celle faite dans [Miq01]. Ensuite nous introduirons les modèles de Kripke, souvent utilisés comme modèle de la logique intuitionniste, et qui intervient dans la méthode de preuve développée dans [Gog94] pour prouver certaines propriétés de UTT. Nous rappelons que c'est cette même méthode que nous réutilisons dans cette thèse. Pour finir, nous fournirons une très brève définition générale des modèles du lambda-calcul.

Lorsque l'on définit un modèle pour décrire la sémantique d'une théorie, il y a deux propriétés incontournables que l'on retrouve : la correction et la complétude. Une théorie T est correcte par rapport à une sémantique S si chaque formule dé-

[1]Nous rappelons que $CC\omega$ est le fragment purement fonctionnel de ECC. ECC étant à la base du développement de LF.

rivable dans T est vraie dans la sémantique S. De même, on parle de complétude d'une théorie T par rapport à une sémantique S, si toute formule vraie de S est dérivable dans T.

1.1.1 Le modèle ensembliste

Dans le modèle ensembliste, chaque concept de la théorie des types est traduit par des notions de la théorie classique des ensembles. Afin de prouver les bonnes propriétés de UTT^r, la construction d'un modèle ensembliste sera nécessaire dans la section 3.2.4.

De manière générale et plutôt intuitive nous avons :

- la relation de typage $M : T$ est traduite par la relation d'appartenance,
$$[|M : T|] = [|M|] \in [|T|]$$

- les fonctions de la théorie des types $\lambda x : T.M_x$ sont traduites par les fonctions de la théorie des ensembles notées $\mapsto$,
$$[|\lambda x : T.M_x|] = x_{\in [|T|]} \mapsto [|M_x|]$$

- l'application de la théorie des types $M(N)$ est traduite par l'application d'une fonction à son argument telle qu'elle est définie dans la théorie des ensembles,
$$[|M(N)|] = [|M|]([|N|])$$

- le produit dépendant de la théorie des types $\Pi x : T.U_x$ est traduit par le produit d'une famille d'ensembles,
$$[|\Pi x : T.U_x|] = \Pi_{x \in [|T|]}[|U_x|]$$

- les propositions sont traduites tout simplement par des booléens, soit une proposition est vraie, soit elle est fausse,

$$[|Prop|] = \{0,1\} \quad \text{où } 0 = \emptyset \text{ et } 1 = \{*\}$$

- la suite des univers prédicatifs $(Type_i)_{i>0}$ est interprétée par une suite d'ensembles $(V_i)_{i>0}$ souvent représentée par une hiérarchie cumulative,

$$[|Type_i|] = V_i$$

Il faut noter que ce modèle interprète toutes les fonctions de la théorie des types en des fonctions de la théorie des ensembles, c'est-à-dire des ensembles de couples argument/résultat. Considérons par exemple la fonction identité sur les entiers, elle sera codée en théorie des types par $\lambda x : Nat.x : Nat \Rightarrow Nat$, tandis que son codage en théorie des ensembles sera $\{(0,0);(1,1);(2,2);\ldots\}$. De la même façon, un type T est transformé en un ensemble contenant les dénotations des termes M de ce type T. En ce qui concerne les propositions, elles sont interprétées par des booléens. Comme en théorie des types, les propositions sont les types de leurs preuves, les propositions sont donc traduites par des ensembles dont les éléments sont les dénotations des preuves de ces propositions. Autrement dit, cette interprétation fait apparaître un contenu extensionnel, pour une théorie des types dont la vocation est de proposer un aspect intensionnel du calcul. En effet deux termes β-convertibles sont systématiquement interprétés par le même ensemble, de même que toutes les propositions vraies auront la même interprétation. On voit donc que cette interprétation s'interesse davantage au résultat du calcul plutôt qu'au calcul lui-même, et qu'elle est de ce fait bien plus apte à prouver des résultats liés à la prouvabilité que des résultats liés à la calculabilité, tels que la normalisation forte. Il est toutefois possible de prouver la normalisation forte d'un calcul comme $CC\omega$ avec des outils de sémantique dénotationnelle. Dans ce cas, il faut enrichir l'interprétation avec des objets syntaxiques (par exemple des candidats de réducibilité) afin de conserver dans ce modèle une trace des étapes du calcul.

1.1.2 Le modèle catégorique

Nous allons très brièvement décrire ici le modèle catégorique qui est un modèle intuitionniste. Contrairement au modèle ensembliste, ce modèle répond de manière bien plus satisfaisante au problème posé par l'étude de la structure intui-

tionniste de $CC\omega$.
Ce modèle est utilisé, rappelons-le dans [Luo94b] pour fournir une interprétation possible du calcul $CC\omega$. Ce modèle se situe dans la catégorie des ω-sets, et possède les caractéristiques suivantes :

- Les types ne sont pas interprétés par des ensembles uniquement, mais par des ensembles munis d'une relation de réalisabilité, appelés ω-sets. Plus formellement, un ω-set est un couple $X = (|X|, \models_X)$ formé par :

  - un ensemble $|X|$ quelconque appelé support de X,

  - une relation surjective $(\models_X) \subset \omega \times |X|$ appelée relation de réalisabilité sur X ;
    (Intuitivement, les réalisateurs d'un objet $x \in |X|$ donné désignent tous les numéros des fonctions récursives partielles susceptibles de représenter l'objet x au sein du type X.)

  - le produit dépendant $\Pi x : X.Y_x$ est interprété par le ω-set produit $\Pi(x \in X; Y_x)$. Pour un ω-set X et $(Y_x)_{x\in|X|}$ une famille de ω-sets, $\Pi(x \in X; Y_x)$ est définie par :

    * $|\Pi x \in X; Y_x| \equiv \{f \in \Pi_{x\in|X|}|Y_x|;$ f a au moins un réalisateur n$\}$

    * $n \models_{\Pi(x\in X;Y_x)} f \equiv \forall p, x \quad p \models_X x \Rightarrow n \cdot p \models_{Y_x} f(x)$ où $n \cdot p$ signifie : "n-ième fonction récursive appliquée à p"

  - les propositions sont interprétées par les PER[2] et les termes de preuves par des fonctions récursives partielles ;

  - l'interprétation des univers prédicatifs suit un schéma très analogue au cas du modèle ensembliste. On considère une suite de grands ensembles emboîtés $(V_i)_{i>0}$ dont chacun des éléments est à lui seul un modèle de la théorie des ensembles. On interprète $Type_i$ par l'ensemble des ω-sets$\in V_i$.

[2]Un PER est une relation d'équivalence partielle sur ω, c'est-à-dire une relation $R \subset \omega \times \omega$ symétrique et transitive.

1.1.3 Les modèles de Kripke

Les modèles de Kripke sont bien connus dans le domaine de la logique intuitionniste. Cependant, il en existe différentes versions dans la littérature. Nous présentons une version générale des modèles de Kripke pour la logique intuitionniste [Cro65].

Un modèle de Kripke est une structure M de la forme $(W, \preceq, h, \models)$ où :

- W est un ensemble non vide de mondes dit univers ;
- $\preceq$ est une relation binaire réflexive et transitive dite d'accessibilité ;
- $h : Variable \rightarrow \wp(W)$[3] est une fonction de valuation, intuitivement si $h(x)$ est l'ensemble des mondes où x est satisfait ; on demande que h soit une fonction dirigée, c'est-à-dire si $w_j \in h(a)$ et $w_i \preceq w_j$ alors $w_j \in h(a)$;
- $\models$ est une relation de forçage ou encore de réalisabilité entre un élément $w_i \in W$ et une proposition φ, $w_i \models \varphi$ se lit φ est satisfait dans w_i.
  - Si φ est une variable x, on a $w_i \models \varphi$ si et seulement si $w_i \in h(x)$.
  - Si φ est une conjonction $\Psi \wedge \Theta$, on a $w_i \models \varphi$ si et seulement si $w_i \models \Psi$ et $w_i \models \Theta$.
  - Si φ est une disjonction $\Psi \vee \Theta$, on a $w_i \models \varphi$ si et seulement si $w_i \models \Psi$ ou $w_i \models \Theta$.
  - Si φ est une implication $\Psi \Rightarrow \Theta$, on a $w_i \models \varphi$ si et seulement si pour tout $w_j \geq w_i$, $w_j \models \Psi$ implique $w_j \models \Theta$.

Dans [MM96], on retrouve une classe particulière de modèles de Kripke pour le lambda-calcul, les "Kripke lambda models". Dans [CG90], une preuve de normalisation pour les théories de constructions est réalisée à l'aide des "Kripke lambda models", où les contextes jouent le rôle des mondes. Ce même principe est appliqué dans la preuve de Goguen [Gog94] pour prouver les propriétés du calcul UTT.

[3]Pour un ensemble W, $\wp(W)$ désigne l'ensemble des parties de W.

1.1.4 λ-modèles

La notion de λ-modèle est la structure sur laquelle repose l'interprétation des termes dans un modèle. En axiomatisant cette structure, on peut l'abstraire de la description des modèles du calcul et ainsi gagner en généralité.
Nous rappelons ici une définition générale concernant la théorie des modèles du lambda-calcul [Bar84].

Définition 1.1 (Valuation)**.** *Soit S un ensemble. Une valuation sur S est une application $\rho : V \to S$ associant un objet $\rho(x) \in S$ à toute variable $x \in V$. L'ensemble de toutes les valuations sur S est noté $\textbf{\textit{Val}}_S$.*

Si ρ est une valuation sur S, x une variable et a un élément de S, on note $\rho; x \leftarrow a$ la valuation construite en associant a à x dans ρ, effaçant ainsi la valeur précédemment associée à x.

Définition 1.2 (λ-modèle)**.** *Un λ-modèle est un triplet $(S, \cdot, [||])$ tel que :*

- *S est un ensemble ayant au moins un élément,*
- *$(\cdot) : S \times S \to S$ est une opération binaire sur S,*
- *$[||] : \Lambda \times \textbf{\textit{Val}}_S \to S$ (où Λ désigne l'ensemble des termes du λ-calcul considéré) est une application associant un objet $[|M|]_\rho \in S$ à n'importe quel terme $M \in \Lambda$ et à n'importe quelle valuation $\rho \in \textbf{\textit{Val}}_S$, qui satisfait les axiomes suivants :*

    1. *$[|x|]_\rho = \rho(x)$,*
    2. *$[|MN|]_\rho = [|M|]_\rho \cdot [|N|]_\rho$,*
    3. *$[|\lambda x.M|]_\rho \cdot a = [|M|]_{\rho;x\leftarrow a}$,*
    4. *Si $[|M|]_{\rho;x\leftarrow a} = [|M'|]_{\rho';x\leftarrow a}$ pour tout $a \in S$, alors $[|\lambda x.M|]_\rho = [|\lambda x.M'|]'_\rho$.*

1.2 Notion de sous-typage

Parmi les récentes avancées dans le domaine de la théorie de types, il faut mentionner la prise en compte de la notion de sous-typage. Il existe plusieurs approches de la notion de sous-typage, la plus traditionnelle des approches traite le sous-typage juste comme une relation entre deux types ($A_0 < A$ dénote que A_0 est un sous-type de A). De ce point de vue, si un terme t est de type A_0, ce même terme t sera également considéré comme étant de type A. Ce fait est illustré par l'utilisation de la règle (subsumption) :

$$\frac{\Gamma \vdash t : A_0 \qquad \Gamma \vdash A_0 < A}{\Gamma \vdash t : A}.$$

Cette approche pose le problème de l'unicité du type et présente une de ses limites pour les théories de types comme celles de Martin-Löf [ML84] ou celles que l'on peut définir dans LF comme UTT^{τ}. En effet dans ces théories, les types de données (en opposition aux propositions) sont généralement des types inductifs dont on distingue parmi leurs éléments les objets canoniques (les éléments construits directement par application des constructeurs). Évidemment, la plupart des opérations définies pour un type inductif s'adresse à ses objets canoniques. Il est possible que la représentation des éléments change quand on passe du sous-type A_0 au type A, et dans le cas des types inductifs l'élément $t : A_0$ traité comme un élément de type A n'aura pas les mêmes propriétés que les éléments de type A qui peuvent se réduire à un objet canonique de A. L'exemple qui suit illustre ce problème.

Exemple 1.3. *Si on considère un type inductif F_3 représentant un type fini (voir section 2.4). $F_3 = ind(\alpha).(1_{F_3} : \alpha, 2_{F_3} : \alpha, 3_{F_3} : \alpha)$, où l'entier 1 est représenté par 1_{F_3}, l'entier 2 par 2_{F_3}, et l'entier 3 par 3_{F_3}.*
On considère également le type inductif Nat, qui représente l'ensemble des entiers naturels, de manière classique : $Nat = ind(\alpha).(0 : \alpha, succ : \alpha \to \alpha)$.
Évidemment F_3 peut être vu comme un sous-type de Nat ($F_3 < Nat$).
Soit la fonction déjà rencontrée dans l'introduction $\lambda x : Nat.(0 + x) : Nat \to Nat$ (où $+$ est définie par une récursion sur le second opérande, où on considère la ι-réduction pour les opérateurs de récursion). Le terme $(\lambda x : Nat.(0+x))(1_{F_3})$ ne se réduit pas à $succ(0)$ contrairement à ce que l'on pourrait penser, car comme vu dans l'introduction, la ι-réduction n'est applicable que lorsque l'argument est 0 ou de la forme $succ(t)$.

Ces observations ont amené la communauté qui travaille sur la théorie des types et ses applications à considérer d'autres approches du sous-typage, notamment le sous-typage coercitif. La notion de coercion est présentée comme une représentation explicite de la transformation du (des éléments du) sous-type vers le (les éléments du) super-type. La relation de sous-typage ($A_0 < A$) est interprétée par l'existence d'un certain terme $c : A_0 \to A$. Le mécanisme de coercion avec certaines restrictions a été implémenté dans les systèmes d'assistance à la preuve Lego [LP92] et Coq [Coq96] par Bailey [Bai96] et Saibi [Sai97] respectivement. Callaghan du Computer Assisted Reasoning Group a également implémenté Plastic un assistant à la preuve qui supporte LF et le sous-typage coercitif.

Dans le developpement de l'approche du sous-typage coercitif, on peut distinguer deux grandes directions. Concernant la première [LMS95, SMMS91, SP94], les termes de coercions possèdent des propriétés particulières qui les distinguent des autres termes dérivables de la théorie. Par exemple dans [LMS95], les coercions sont imposées par les règles d'inférences du système, et elles sont définies comme étant "presque" la fonction d'identité. Ces coercions sont caractérisées formellement par des termes, qui après effacement des informations de type, se η-réduisent toutes à l'identité. Dans la seconde approche [Luo97c, JLS98], il n'y a pas de restrictions particulières sur le terme de coercion entre un type et son sous-type. Les coercions sont n'importe quelle application entre deux types, incluant celles définies par l'utilisateur. Non seulement sont acceptées comme coercions les applications particulières des approches vues plus haut, mais sont également acceptées toutes les applications arbitraires vérifiant la condition de cohérence qui stipule qu'il est impossible d'avoir deux coercions différentes $c_1, c_2 : A_0 \to A$ pour le même sous-type A_0 de A. Le rôle d'un terme de coercion c est illustré par la règle suivante :

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k_0 : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_0) = f(c(k_0)) : [c(k_0)/x]K'} \text{(CD)} .$$

Cette règle stipule que si f est une fonction ayant comme domaine K, k_0 un objet de K_0, et c une coercion de K_0 vers K, alors le terme $f(k_0)$ est bien typé et est intensionnellement égal à $f(c(k_0))$. Intuitivement, on peut voir f comme un contexte qui requiert un objet de K, alors l'argument k_0 dans le contexte f représente son image par la coercion, $c(k_0)$. Par conséquent on peut utiliser $f(k_0)$ comme une abréviation de $f(c(k_0))$.

Ce concept simple, formulé dans un cadre comme LF devient très puissant.

Chapitre 2

Le système UTT^r

Dans ce chapitre, nous présentons le calcul UTT^r. Ce système est une extension de la version du calcul UTT considérée dans [Gog94][1], par l'ajout d'une règle de jugement d'égalité (voir section 2.4). Le calcul UTT^r est formulé dans le "Logical Framework" LF [Luo94b], un système formel qui a pour vocation d'être utilisé comme un cadre pour définir des théories de types. LF est une théorie simple possédant des types fonctionnels, des types dépendants, et dans laquelle on définit de nouvelles théories par ajout de constantes et de règles d'égalité.

UTT^r comprend un univers imprédicatif des propositions qui est distinct de la collection des types. On inclut le type des propositions et les types des preuves de chaque proposition dans cet univers. Nous retrouvons également un univers prédicatif des types. Cet univers inclut les types inductifs (qui sont importants pour la programmation et les mathématiques), comme les entiers naturels, le type produit, les listes etc. UTT^r fournit un excellent environnement pour étudier les spécifications et la vérification de programmes. Luo discute cette approche dans [Luo93a, Luo94b].

2.1 "Logical Framework" LF

Les termes de LF sont de la forme suivante : **Type**, $El(A)$, $(x : K)K'$, $[x : K]k'$, $f(k)$, où les occurrences libres de la variable x sont liées par les opérateurs $(x : K)$ et $[x : K]$ respectivement.

[1]Contrairement à la version de [Luo94b], cette version de UTT ne contient qu'un seul univers prédicatifs.

Il y a cinq formes de jugement dans LF :

- $\Gamma \vdash$ **valid**, signifie Γ est un contexte valide ;
- $\Gamma \vdash K\textbf{kind}$, signifie que K est une sorte,
- $\Gamma \vdash k : K$ signifie que k est un objet de sorte K,
- $\Gamma \vdash k = k' : K$ signifie que k et k' sont des objets égaux de sorte K,
- $\Gamma \vdash K = K'$ signifie que K et K' sont deux sortes égales.

Les règles d'inférence de LF sont les suivantes :

- **Contexte et assomption**

$$\frac{}{<>\vdash \textbf{valid}}\ (1.1) \qquad \frac{\Gamma \vdash K\textbf{kind} \quad x \notin FV(\Gamma)}{\Gamma, x : K \vdash \textbf{valid}}\ (1.2) \qquad \frac{\Gamma, x : K, \Gamma' \vdash \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K}\ (1.3)$$

$$\frac{\Gamma_1, \Gamma_2 \vdash J \qquad \Gamma_1, \Gamma_3 \vdash \textbf{valid}}{\Gamma_1, \Gamma_3, \Gamma_2 \vdash J}\ (\text{wkn}) \qquad (FV(\Gamma_2) \cap FV(\Gamma_3) = \emptyset)$$

- **Égalité générale**

$$\frac{\Gamma \vdash K\textbf{kind}}{\Gamma \vdash K = K}\ (2.1) \qquad \frac{\Gamma \vdash K = K'}{\Gamma \vdash K' = K}\ (2.2) \qquad \frac{\Gamma \vdash K = K' \qquad \Gamma \vdash K' = K''}{\Gamma \vdash K = K''}\ (2.3)$$

$$\frac{\Gamma \vdash k : K}{\Gamma \vdash k = k : K}\ (2.4) \qquad \frac{\Gamma \vdash k = k' : K}{\Gamma \vdash k' = k : K}\ (2.5) \qquad \frac{\Gamma \vdash k = k' : K \qquad \Gamma \vdash k' = k'' : K}{\Gamma \vdash k = k'' : K}\ (2.6)$$

- **Retypage**

$$\frac{\Gamma \vdash k : K \qquad \Gamma \vdash K = K'}{\Gamma \vdash k : K'}\ (3.1) \qquad \frac{\Gamma \vdash k = k' : K \qquad \Gamma \vdash K = K'}{\Gamma \vdash k = k' : K'}\ (3.2)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash J \qquad \Gamma \vdash K = K'}{\Gamma, x : K', \Gamma' \vdash J}\ (3.3)$$

- **La sorte Type**

$$\frac{\Gamma \vdash \textbf{valid}}{\Gamma \vdash \textbf{Type kind}}\ (4.1) \qquad \frac{\Gamma \vdash A : \textbf{Type}}{\Gamma \vdash El(A)\textbf{kind}}\ (4.2) \qquad \frac{\Gamma \vdash A = B : \textbf{Type}}{\Gamma \vdash El(A) = El(B)\textbf{kind}}\ (4.3)$$

- **Produits dépendants**

$$\frac{\Gamma, x : K \vdash K'\textbf{kind}}{\Gamma \vdash (x : K)K'\textbf{kind}}\ (5.1) \qquad \frac{\Gamma, x : K_1 \vdash K_1' = K_2' \qquad \Gamma \vdash K_1 = K_2}{\Gamma \vdash (x : K_1)K_1' = (x : K_2)K_2'}\ (5.2)$$

$$\frac{\Gamma, x : K \vdash k : K'}{\Gamma \vdash [x : K]k : (x : K)K'}\ (5.3) \qquad \frac{\Gamma, x : K_1 \vdash k_1 = k_2 : K \qquad \Gamma \vdash K_1 = K_2}{\Gamma \vdash [x : K_1]k_1 = [x : K_2]k_2 : (x : K_1)K}\ (5.4)$$

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k : K}{\Gamma \vdash f(k) : [k/x]K'}\ (5.5) \qquad \frac{\Gamma \vdash f = f' : (x : K)K' \qquad \Gamma \vdash k_1 = k_2 : K}{\Gamma \vdash f(k_1) = f'(k_2) : [k_1/x]K'}\ (5.6)$$

$$\frac{\Gamma, x : K \vdash k' : K' \qquad \Gamma \vdash k : K}{\Gamma \vdash ([x : K]k')(k) = [k/x]k' : [k/x]K'}\ (5.7) \qquad \frac{\Gamma \vdash f : (x : K)K' \quad x \notin FV(\Gamma)}{\Gamma \vdash [x : K]f(x) = f : (x : K)K'}\ (5.8)$$

- **Substitutions**

$$\frac{\Gamma, x : K, \Gamma' \vdash \textbf{valid} \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash \textbf{valid}}\ (6.1) \qquad \frac{\Gamma, x : K, \Gamma' \vdash K'\textbf{kind} \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]K'\textbf{kind}}\ (6.2)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash k' : K' \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]k' : [k/x]K'\textbf{kind}}\ (6.3) \qquad \frac{\Gamma, x : K, \Gamma' \vdash K' = K'' \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]K' = [k/x]K''}\ (6.4)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash k' = k'' : K' \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]k' = [k/x]k'' : [k/x]K'}\ (6.5)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash K'\textbf{kind} \qquad \Gamma \vdash k_1 = k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]K' = [k_2/x]K'}\ (6.6)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash k' : K' \qquad \Gamma \vdash k_1 = k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]k' = [k_2/x]k' : [k_1/x]K'}\ (6.7)$$

Définition 2.1 (Contexte)**.** *Une séquence $x_1 : A_1, \dots, x_n : A_n$ de couples formés d'une variable x_i et d'une sorte A_i est un pré-contexte.*
Le domaine d'un pré-contexte Γ, noté $dom(\Gamma)$, est l'ensemble des variables $\{x_1, \dots, x_n\}$.
*Un pré-contexte Γ, tel que le jugement $\Gamma \vdash$ **valid** est dérivable, est un contexte.*

Notation

- On note $(K)K'$ pour désigner la sorte $(x : K)K'$ quand x n'apparaît pas parmi les variables libres de K'.
- On note également $f(k_1, \dots, k_n)$ pour désigner $f(k_1) \dots (k_n)$.

2.1.1 Spécifier une théorie de types dans LF

En général, la spécification d'une théorie de types dans LF consiste en une collection de déclarations de nouvelles constantes et de règles de calcul. Formellement, déclarer une nouvelle constante k de sorte K, c'est introduire la règle d'inférence suivante dans la théorie spécifiée :

$$\frac{\Gamma \vdash \textbf{valid}}{\Gamma \vdash k : K}$$

Ici k ne contient pas de variable et K ne contient pas de variable libre. Déclarer que $k = k'$ aux conditions $k_i : K_i$ $(i = 1, \dots, n)$ revient à enrichir la théorie spécifiée de la règle d'inférence :

$$\frac{\Gamma \vdash k_i : K_i \quad (i = 1, \dots, n) \qquad \Gamma \vdash k : K \qquad \Gamma \vdash k' : K}{\Gamma \vdash k = k' : K}$$

Exemple 2.2. *On introduit le type Π en déclarant les constantes :*

$$\Pi : (A : \textbf{\textit{Type}})((A)\textbf{\textit{Type}})\textbf{\textit{Type}}$$

$$\lambda : (A : \textbf{\textit{Type}})(B : (A)\textbf{\textit{Type}})((x : A)B(x))\Pi(A, B)$$

$$\boldsymbol{E}_\Pi : (A : \textbf{\textit{Type}})(B : (A)\textbf{\textit{Type}})(C : (\Pi(A, B))\textbf{\textit{Type}}((g : (x : A)B(x))C(\lambda(A, B, g)))(z : \Pi(A, B))C(z)$$

De même que l'égalité calculatoire suivante :

$$\boldsymbol{E}_{\Pi}(A, B, C, f, \lambda(A, B, g)) = f(g) : C(\lambda(A, B, g)).$$

avec A : ***Type****,* $B : (A)$***Type****,* $C : (\Pi(A, B))$***Type****,* $f : ((g : (x : A)B(x))C(\lambda(A, B, g))$ *et* $g : (x : A)B(x)$*.*

Notons que nous omettons ici l'opérateur El*.*

2.1.2 Schéma pour les types inductifs

Le système UTT^r contient des types inductifs dont l'introduction repose sur la notion de schéma inductif. Nous allons considérer ici les opérateurs et constantes de LF permettant d'introduire de manière générale des types inductifs.

Définition 2.3 (Petite sorte)**.** *On dit qu'une sorte A est une petite sorte (small kind) si*

- $A \equiv El(M)$ *ou*
- $A \equiv (x : A_1)A_2$*, où* A_1 *et* A_2 *sont des petites sortes.*

Définition 2.4 (Famille de schémas)**.** *Soit* Γ *un contexte valide et* X *une variable.*

- *Une sorte* Φ *est un opérateur strictement positif dans* Γ *par rapport à* X*,* notation $POS_{\Gamma;X}(\Phi)$*, si* Γ, X : ***Type*** $\vdash \Phi$***kind*** *et*
  - $\Phi \equiv El(X)$ *ou*
  - $(x : A)\Phi_0$*, où A est une petite sorte et* $POS_{\Gamma;X}(\Phi_0)$*.*
- *Une sorte* Θ *est un schéma inductif par rapport à* X*,* notation $SCH_{\Gamma;X}(\Theta)$*, si*
  - $\Theta \equiv El(X)$*,*
  - $(x : A)\Theta_0$*, où A est une petite sorte et* $SCH_{\Gamma;X}(\Theta_0)$*, ou*

– $(\Phi)\Theta_0$, *où* $POS_{\Gamma;X}(\Phi)$ *et* $SCH_{\Gamma;X}(\Theta_0)$.

- *Une séquence finie de sortes* $\langle\Theta_1,\ldots,\Theta_n\rangle$, *notée* $\bar{\Theta}$, *est une famille de schémas dans* Γ *par rapport à* X, notation $SCH_{\Gamma;X}(\Theta_1,\ldots,\Theta_n)$, *si* $SCH_{\Gamma;X}(\Theta_i)$ *pour* $1 \leq i \leq n$.

Notation

- *Par la suite, nous prendrons l'habitude d'omettre l'opérateur* El *lorsque nous considérerons les opérateurs et constantes relatifs aux types inductifs ainsi que certaines constantes déclarées.*

- *Nous notons* $\Phi(M)$ *pour* $[M/X]\Phi$ *si* $POS_{\Gamma;X}(\Phi)$, *pareillement pour* Θ *telle que* $SCH_{\Gamma;X}(\Theta)$.

- *Si* Θ *est un schéma inductif par rapport à* X *alors nous notons* $ARITY_X(\Theta)$ *la séquence* $\Phi_1,\ldots,\Phi_m$ *des opérateurs strictement positifs de* Θ *dans* Γ *par rapport à* X.

- *Nous utilisons la notation* $F[x_1,\ldots,x_n] =_{df} M$ *pour présenter des définitions avec des variables* $x_1,\ldots,x_n$. *La notation* $F[N_1,\ldots,N_n]$ *représente donc le terme ou la sorte* $[N_1,\ldots,N_n/x_1,\ldots,x_n]M$.

Définition 2.5.

- *présumons que* $POS_{\Gamma;X}(\Phi)$, *où* $\Phi \equiv (x_1 : A_1)\ldots(x_n : A_n)X$, *et supposons que* $\Gamma \vdash A : \textbf{\textit{Type}}$, $\Gamma \vdash C : (A)\textbf{\textit{Type}}$ *et* $\Gamma \vdash z : \Phi(A)$. *Nous définissons :*
$$\Phi^\circ[A,C,z] =_{df} (x_1 : A_1)\ldots(x_n : A_n)C(z(x_1,\ldots,x_n))$$

- *Supposons que* $SCH_{\Gamma;X}(\Theta)$ *où* $\Theta \equiv (x_1 : M_1)\ldots(x_n : M_n)X$, *soit* A, C *et* z *comme définis précédemment. Nous définissons :*
$$\Theta^\circ[A,C,z] =_{df} (x_1 : M_1(A))\ldots(x_n : M_n(A))(\Phi^\circ_{i_1}[A,C,x_{i_1}])\ldots(\Phi^\circ_{i_k}[A,C,x_{i_k}])C(z(x_1,\ldots,x_n))$$

- *Supposons que* $POS_{\Gamma;X}(\Phi)$, *où* $\Phi \equiv (x_1 : A_1)\ldots(x_n : A_n)X$, *soit* A, C *et* z *comme définis précédemment, et soit* $\Gamma \vdash f : (x : A)C(x)$. *Nous* *définissons :*

$$\Phi^\natural[A, C, f, z] =_{df} [x_1 : A_1] \ldots [x_n : A_n] f(z(x_1, \ldots, x_n))$$

Nous avons les règles d'introduction suivantes, pour les constantes associées aux types inductifs. Nous considérons ces règles relativement à un contexte valide Γ *et* $\bar{\Theta}$ *une famille de schémas dans* Γ *par rapport à* X *:*

$$M^X[\bar{\Theta}] : \quad \textbf{\textit{Type}}$$

$$\iota_i^X[\bar{\Theta}] : \quad \Theta_i(M^X[\bar{\Theta}]) \quad (1 \leq i \leq n)$$

$$E^X[\bar{\Theta}] : \quad (C : (M^X[\bar{\Theta}])\textbf{\textit{Type}})$$

$$(f_1 : \Theta_1^\circ[M^X[\bar{\Theta}], C, \iota_1^X[\bar{\Theta}]])$$

$$\ldots$$

$$(f_n : \Theta_n^\circ[M^X[\bar{\Theta}], C, \iota_n^X[\bar{\Theta}]])$$

$$(z : M^X[\bar{\Theta}])C(z)$$

Nous avons les règles d'égalité associées :

$$\begin{gathered} E^X[\bar{\Theta}](C, \bar{f}, \iota_i^X[\bar{\Theta}](\bar{a})) = \\ f_i(\bar{a}, \Phi_1^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_1], \ldots, \Phi_k^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_k]) : \\ C(\iota_i^X[\bar{\Theta}](\bar{a})) \end{gathered}$$

pour $1 \leq i \leq n$*, avec* $\bar{f}$ *qui représente* $f_1, \ldots, f_n$*,* $\bar{a}$ *qui représente* $a_1, \ldots, a_m$ *et* $ARITY_X(\Theta_i) \equiv \Phi_1, \ldots, \Phi_k$*.*

Afin de mieux appréhender la définition ci-dessus, considérons l'exemple simple qui déclare un type inductif représentant les entiers naturels.

Exemple 2.6. *Nous déclarons ici le types des entiers naturels :*

$$N : \quad \textbf{\textit{Type}}$$

$$0 : \quad N$$

$$succ : \quad (N)N$$

$$Rec_N : \quad (C : (N)\textbf{\textit{Type}})$$

$$(c : C(0))$$
$$(f : (x : N)(C(x))C(succ(x)))$$
$$(n : N)C(n)$$

Nous avons les règles d'égalité associées :

$$Rec_N(C, c, f, 0) = c : C(0)$$
$$Rec_N(C, c, f, succ(n)) = f(n, Rec_N(C, c, f, n)) : C(succ(n))$$

Afin de montrer que cette déclaration du type des entiers naturels correspond bien à celle d'un type inductif dans le sens de la définition 2.5, nous établissons les liens suivants :

- *le type N est défini par $N =_{df} M^X[\bar{\Theta}_N]$,*
- *où $\bar{\Theta}_N \equiv \langle X, (X)X \rangle$,*
- *le premier constructeur est $0 =_{df} \iota_1^X[\bar{\Theta}_N]$,*
- *le second constructeur est $succ =_{df} \iota_2^X[\bar{\Theta}_N]$,*
- *l'opérateur de récursion est $Rec_n \equiv E^X[\bar{\Theta}_N]$.*

Définition 2.7 (Expressions constantes)**.** *Soit $\bar{\Theta}$ une famille de schémas inductifs dans Γ par rapport à X. Une expression constante (constant expression) par rapport à $\bar{\Theta}$ est une expression de la forme $\kappa^X[\bar{\Theta}]$, où les occurrences libres de X deviennent liées (par l'opérateur [_]). κ^X est la lettre constante (constant letter) de l'expression constante $\kappa^X[\bar{\Theta}]$.*

Il faut considérer également la règle d'égalité suivante relative aux constantes associées aux schémas inductifs :

$$(\kappa^X\text{-EQ})\ \frac{\begin{array}{c}SCH_{\Gamma,X}(\bar{\Theta}) \quad SCH_{\Gamma,X}(\bar{\Theta}') \qquad \Gamma \vdash \kappa^X[\bar{\Theta}] : K \quad \Gamma \vdash \kappa^X[\bar{\Theta}'] : K \\ \Gamma, X : \textbf{Type} \vdash \bar{\Theta}_i = \bar{\Theta}'_i \quad (i = 1, \ldots, n)\end{array}}{\Gamma \vdash \kappa^X[\bar{\Theta}] = \kappa^X[\bar{\Theta}'] : K}$$

2.2 Un univers imprédicatif des propositions

La logique interne de UTT^r consiste en univers $Prop$ des propositions et le type de leurs preuves. Ils sont introduits en déclarant les constantes suivantes :

$$Prop : \mathbf{Type}$$

$$\mathbf{Prf} : (Prop)\mathbf{Type}$$

$$\forall : (A : \mathbf{Type})((A)Prop)Prop$$

$$\Lambda : (A : \mathbf{Type})(P : (A)Prop)((x : A)\mathbf{Prf}(P(x)))\mathbf{Prf}(\forall(A, P))$$

$$\mathbf{E}_\forall : (A : \mathbf{Type})(P : (A)Prop)(R : \mathbf{Prf}(\forall(A, P)))Prop)((g : (x : A)\mathbf{Prf}(P(x)))\mathbf{Prf}(R(\Lambda(A, P, g))))(z : \mathbf{Prf}(\forall(A, P)))\mathbf{Prf}(R(z))$$

On a l'égalité calculatoire suivante :

$$\mathbf{E}_\forall(A, P, R, f, \Lambda(A, P, g)) = f(g) : \mathbf{Prf}(R(\Lambda(A, Pg))).$$

L'univers des propositions est imprédicatif dans le sens où le quantificateur universel parcourt des types arbitraires. L'expression $\forall(Prop, [X : Prop]Prf(X))$ quantifie toutes les propositions, et est elle-même une proposition.

2.3 Univers prédicatif

L'univers prédicatif contient les noms des types et permet de quantifier sur les types et d'écrire des fonctions qui retournent des types. Les déclarations qui définissent l'univers prédicatif sont :

$$\mathbf{U} : \mathbf{Type}$$

$$\mathbf{T} : (\mathbf{U})\mathbf{Type}$$

$$prop : \mathbf{U}$$

$$prf : (Prop)\mathbf{U}$$

Les règles d'égalité associées sont les suivantes :

$$\frac{\Gamma \vdash \textbf{valid}}{\Gamma \vdash \mathbf{T}(prop) = Prop : \textbf{Type}} \text{ (T-prop)} \qquad \frac{\Gamma \vdash P : Prop}{\Gamma \vdash \mathbf{T}(prf(P)) = \textbf{Prf}(P) : \textbf{Type}} \text{ (T-prf)}$$

Afin d'introduire les types inductifs dans l'univers, nous avons besoin d'un opérateur auxiliaire qui collecte les types intervenant dans les schémas inductifs. Ces opérations sont définies uniquement sur la syntaxe des schémas.

Définition 2.8 ($TYPES_\Gamma(\bar{\Theta})$)**.** *Soit Γ un pré-contexte et soit A, Φ et Θ une petite sorte, un opérateur strictement positif par rapport à X et un schéma inductif par rapport à X. Les ensembles $TYPES_\Gamma(A)$, $TYPES_\Gamma(\Phi)$ et $TYPES_\Gamma(\Theta)$ sont définis de la manière suivante :*

$$TYPES_\Gamma(A) =_{df} \begin{cases} \{(\Gamma, A)\} & \textit{si } A \textit{ est un type} \\ TYPES_\Gamma(A_1) \cup TYPES_{\Gamma,x:A_1}(A_2) & \textit{si } A \equiv (x : A_1)A_2 \end{cases}$$

$$TYPES_\Gamma(\Phi) =_{df} \begin{cases} \emptyset & \textit{si } \Phi \equiv X \\ TYPES_\Gamma(A_1) \cup TYPES_{\Gamma,x:A_1}(\Phi_0) & \textit{si } \Phi \equiv (x : A_1)\Phi_0 \end{cases}$$

$$TYPES_\Gamma(\Theta) =_{df} \begin{cases} \emptyset & \textit{si } \Theta \equiv X \\ TYPES_\Gamma(A_1) \cup TYPES_{\Gamma,x:A_1}(\Theta_0) & \textit{si } \Theta \equiv (x : A_1)\Theta_0 \\ TYPES_\Gamma(\Phi) \cup TYPES_\Gamma(\Theta_0) & \textit{si } \Theta \equiv (\Phi)\Theta_0 \end{cases}$$

Pour $\bar{\Theta} \equiv \Theta_1, \ldots, \Theta_n$, on a $TYPES_\Gamma(\bar{\Theta}) \equiv \bigcup_{1 \le i \le n} TYPES_\Gamma(\Theta_i)$

Supposons que $SCH_{\Gamma;X}(\bar{\Theta})$, et présumons qu'il existe un schéma $\bar{\Theta}'$ tel que $SCH_{\Gamma;X}(\bar{\Theta}')$ et Γ, X : **Type** $\vdash \Theta_i = \Theta'_i$ pour $1 \le i \le n$, et que pour chaque $(\Gamma', A) \in TYPES_\Gamma(\bar{\Theta}')$ il existe un terme a tel que $\mathbf{T}(a) \equiv A$; alors $\mu^X[\bar{\Theta}]$ est un élément de l'univers :

$$\Gamma \vdash \mu^X[\bar{\Theta}] : \mathbf{U} \quad \text{et} \quad \Gamma \vdash \mathbf{T}(\mu^X[\bar{\Theta}]) = M^X[\bar{\Theta}] : \textbf{Type}$$

2.4 La ϑ-réduction

Nous définissons une nouvelle réduction pour le calcul (voir définition 3.4). Nous nous intéressons ici à la représentation d'ensembles finis comme types inductifs.

Notons $[n]$ l'ensemble $\{i \in \mathbb{N}/1 \leq i \leq n\}$. On définit le type $F_n \equiv M^X[\bar{\Theta}_n]$, où $\bar{\Theta}_n \equiv < \underbrace{X, \ldots, X}_{n fois} >$. Rappelons que les termes $\iota_i^X[\bar{\Theta}_n] : F_n$ représentent les objets canoniques de F_n et symbolisent les éléments de l'ensemble $[n]$.

A toute permutation $\sigma : [n] \rightarrow [m]$, correspond le terme $E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m$.

Soit la relation de récriture ϑ telle que
$E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)) \quad \vartheta \quad E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k)$,
avec :
$C \equiv [y : F_m]A : (y : F_m)$**Type**,
$C_1 \equiv [z : F_n]F_m : (z : F_n)$**Type** et
$C_2 \equiv [x : F_n]A : (x : F_n)$**Type** ;
et la ϑ-réduction sa fermeture contextuelle.

Nous rajoutons la ϑ-réduction au calcul sous forme de jugement d'égalité :

$$(\vartheta\text{-eq})\ \frac{\Gamma \vdash E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma \vdash E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m \qquad \Gamma \vdash k : F_n}{\Gamma \vdash E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)) = E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A}$$

Comme nous l'avons précisé dans l'introduction, la règle (ϑ-eq) permet d'introduire sous forme d'égalité intensionnelle une égalité extensionnelle sur les fonctions entre types finis.

Exemple 2.9. *Prenons les ensembles* $[2]$, $[3]$, *et* $[4]$ *ainsi que les types finis correspondants* F_2, F_3 *et* F_4. *Soit* $\sigma : [2] \rightarrow [3]$ *une permutation telle que pour tout* $t \in [2] \quad \sigma(t) = t$, *et les trois fonctions :*
$g \equiv E[\bar{\Theta}_3]([y : F_3]F_4, 1_{F_4}, 2_{F_4}, 3_{F_4}) : (F_3)F_4$,
$f \equiv E[\bar{\Theta}_2]([z : F_2]F_3, 1_{F_3}, 2_{F_3}) : (F_2)F_3$ *et*
$h \equiv E[\bar{\Theta}_2]([x : F_2]F_4, 1_{F_4}, 2_{F_4}) : (F_2)F_4$.
Situation illustrée par le diagramme suivant :

$$\begin{tikzcd} F_2 \arrow[r, "f"] \arrow[dr, "h"'] & F_3 \arrow[d, "g"] \\ & F_4 \end{tikzcd}$$

La ϑ-réduction nous permet de réduire directement le terme $g \circ f(x)$ en $h(x)$, que x soit un objet canonique ou pas de F_2. En effet, en l'absence de la ϑ-réduction nous devrions employer la réduction relative à l'opérateur de récursion de $E[\bar{\Theta}_2]$, qui ne s'applique qu'aux objets canoniques de F_2. Plus concrètement, la règle (ϑ-eq) permet dorénavant de dériver le jugement :
$\Gamma \vdash E[\bar{\Theta}_3]([y : F_3]F_4, 1_{F_4}, 2_{F_4}, 3_{F_4})E[\bar{\Theta}_2](([z : F_2]F_3, 1_{F_3}, 2_{F_3})(k)) = E[\bar{\Theta}_2]([x : F_2]F_4, 1_{F_4}, 2_{F_4})(k) : F_4$ *même quand k est une variable.*

Il faut noter que si nous considérons trois types finis F_n, F_m et F_p, aux permutations $g : [m] \to [p]$ et $f : [n] \to [m]$ correspondent les termes
$E[\bar{\Theta}_m]([y : F_m]F_p, \iota^X_{g(1)}[\bar{\Theta}_p], ..., \iota^X_{g(m)}[\bar{\Theta}_p]) : (F_m)F_p$ et
$E[\bar{\Theta}_n]([z : F_n]F_m, \iota^X_{f(1)}[\bar{\Theta}_m], ..., \iota^X_{f(n)}[\bar{\Theta}_m] : (F_n)F_m$ respectivement. Par application de la ϑ-réduction, on a :
$E[\bar{\Theta}_m]([y : F_m]F_p, \iota^X_{g(1)}[\bar{\Theta}_p], ..., \iota^X_{g(m)}[\bar{\Theta}_p])(E[\bar{\Theta}_n]([z : F_n]F_m, \iota^X_{f(1)}[\bar{\Theta}_m], ..., \iota^X_{f(n)}[\bar{\Theta}_m])(k)) \quad \vartheta$
$E[\bar{\Theta}_n]([x : F_n]F_p, \iota^X_{g \circ f(1)}[\bar{\Theta}_p], ..., \iota^X_{g \circ f(n)}[\bar{\Theta}_p])(k)$.
Ce cas particulier n'est autre que la réduction sur les fonctions entre types finis, introduite par Soloviev et Chemouil dans [SC03] pour un λ-calcul simplement typé avec types inductifs.

Chapitre 3

Propriétés du système UTT^r

Dans ce chapitre, notre objectif est de montrer que certaines propriétés, comme la consistance, la terminaison des calculs, sont vérifiées pour le système UTT^r. Or, nous rappelons que UTT^r est le système UTT enrichi d'une nouvelle règle d'égalité concernant les types finis, à savoir la règle (ϑ-eq). Healdene Goguen a démontré lors de sa thèse [Gog94], sous la direction Rod Burstall et Zhaohui Luo, que le calcul UTT possède certaines propriétés dont la normalisation forte, la préservation du type et la confluence. Goguen, afin de réaliser sa preuve, a développé une sémantique opérationnelle typée pour UTT qu'il nomme UTT^S. Pour construire notre preuve, nous allons donc adapter la technique employée pour prouver les propriétés de normalisation forte, de confluence et de préservation du type du système UTT.

Les réductions non typées fournissent une sémantique opérationnelle naturelle, c'est-à-dire une description de l'aspect calculatoire pour les théories des types. Les lemmes de normalisation forte nous apprennent qu'une telle sémantique est correcte. Cependant, ces réductions ne prennent pas en compte les informations sur les types et ne nous donnent pas d'informations sur les formes canoniques des termes. Goguen introduit donc la sémantique opérationnelle typée qui définit une réduction vers la forme normale pour les termes qui sont bien typés dans la théorie. Une fois la sémantique opérationnelle typée UTT^S définie, Goguen prouve que ce système vérifie certaines propriétés comme la normalisation forte, la préservation du type et la confluence puis il transfère ces propriétés au calcul UTT en démontrant que UTT est correcte par rapport à UTT^S. Nous suivrons dans notre démonstration la même démarche.

3.1 Plan de la partie technique

1. **Le système** UTT^{rS}
   Dans les sections 3.2.1 et 3.2.2, après avoir introduit quelques notions métathéoriques, nous définissons notre système UTT^{rS} qui représente la sémantique opérationnelle typée du système UTT^{r}. Pour cela on rajoute au calcul UTT^{S} la règle suivante qui décrit notre nouvelle réduction avec en plus des informations concernant le typage.

$$
(\text{W-}\vartheta)\ \frac{\begin{array}{c}\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m) \xrightarrow{nf} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \\ \Gamma \vdash^S E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) \xrightarrow{nf} E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m \\ \Gamma \vdash^S k \xrightarrow{nf} v : F_n \quad v \in FV(\Gamma)\end{array}}{\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)) \xrightarrow{wh} E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A}
$$

   Cette règle décrit fidèlement la nouvelle réduction ϑ concernant les types finis que nous considérons dans cette thèse. Elle est bien sûr directement liée à la règle d'égalité rajoutée à UTT^{r}. Lors de l'élaboration de cette règle, nous avions comme impératif, mise à part la description fidèle de la réduction ϑ, de retrouver évidemment, dans UTT^{rS}, les mêmes propriétés que UTT^{S} . Le principal enjeu a été d'éviter tout conflit entre la nouvelle règle (W-ϑ) et toute autre règle, dans le but de préserver les propriétés décrites dans le lemme 3.13 (Génération) qui stipule que tout jugement dérivable dans UTT^{rS} n'admet qu'une et une seule dérivation.

   Dans la section 3.2.3, la preuve est faite que les propriétés de normalisation forte, de confluence et de préservation du type sont vérifiées pour le système UTT^{rS}. Ces preuves sont réalisées au moyen de la définition des prédicats $S^C_\Gamma(A)$ et $S^{P,A}_\Gamma(M)$, où Γ est un contexte, C et A sont des sortes, P et M sont des termes. La définition de ce dernier prédicat est très fortement liée à la notion de normalisation forte et contient en plus des informations sur le type. Le résultat le plus significatif est le lemme 3.32 (Réduction). C'est ce dernier qui va nous permettre d'affirmer que les propriétés que l'on cherche à prouver sont vérifiées pour UTT^{rS}. Par soucis de clarté, nous avons complètement revu la structure de la preuve du lemme 3.32 par rapport à la preuve originale réalisée pour le système UTT^{S}. Pour se faire, l'ajout du lemme 3.31 et un autre principe d'induction ont été nécessaires.

2. **Construction d'un modèle ensembliste classique pour UTT^r**
Dans la section 3.2.4, nous proposons une sémantique de UTT^r dans la théorie classique des ensembles suivant la preuve de Goguen (voir le chapitre A.2 de l'annexe). Dans ce modèle, les types sont interprétés par des ensembles de leurs éléments, à l'exception des propositions qui sont uniquement vraies ou fausses. Les fonctions sont des fonctions ensemblistes, les entiers naturels restent des entiers naturels et en général les types inductifs sont interprétés comme étant les plus petits points fixes de leurs équations inductives. L'importance de ce modèle ensembliste est qu'il va justifier un autre principe d'induction sur les types qui ne pouvait pas être justifié syntaxiquement, utile pour la preuve de la section 3.2.5. Ce principe d'induction en question est décrit dans l'annexe au chapitre A.4.

3. UTT^r **correcte par rapport à** UTT^{rS}
Une fois les propriétés de normalisation forte, de confluence et de préservation du type pour UTT^{rS} démontrées, il nous reste à transférer ces résultats vers le système qui nous interesse, c'est-à-dire UTT^r. Dans la section 3.2.5 nous fournissons la preuve que UTT^r est correcte par rapport à UTT^{rS} en suivant sans modification majeure la démarche de Goguen. Nous donnons ici une description des points importants de la preuve sans entrer dans les détails techniques.

   Les preuves de normalisation fortes et de correction sont en général réalisées par la construction de modèles. Nous élaborons un modèle inspiré des modèles de Kripke. Notre première préoccupation dans la construction d'un modèle est le domaine sémantique de discussion. Dans la section 3.2.5.1, les objets sémantiques sont définis pour la preuve de correction. Comme nous prouvons la correction de UTT^r par rapport à UTT^{rS}, les premiers composants des objets sémantiques sont les termes dérivables dans UTT^{rS}. Il est montré que, relativement à n'importe quelle substitution, tous les termes typables dans UTT^r font partie de l'interprétation de leurs sortes, où les sortes sont interprétées comme étant les ensembles de termes bien typés dans UTT^{rS}. Les secondes composantes des objets sémantiques sont les ensembles valués, qui ont pour but de capturer le comportement ensembliste des éléments de chaque type.

   Une fois les objets sémantiques définis, la section 3.2.5.2 poursuit en proposant une interprétation partielle des termes par des objets sémantiques et des sortes par des ensembles d'objets sémantiques. Le but de notre démarche étant de démontrer certaines propriétés par rapport à l'ajout d'une réduction sur les types finis, nous définissons également dans cette section l'interprétation dans le cas particulier des types finis. Finalement à la section 3.2.5.3, on effectue la preuve finale de correction par induction sur les dérivations de UTT^r.

Nous rappelons que dans ce chapitre, nous effectuons la preuve que le système UTT^r vérifie les propriétés de normalisation forte, de préservation du type et de confluence. Nous suivons sans modification majeure la preuve réalisée par healfdene Goguen, lors de sa thèse sous la direction Rod Burstall et Zhaohui Luo, pour prouver les mêmes propriétés de normalisation forte, de préservation du type et de confluence pour le système UTT. Par conséquent, nous présenterons dans notre preuve les modifications portées à la preuve initiale et n'introduirons que les notions et propriétés nécessaires à la compréhension de notre travail. Pour une meilleure compréhension des étapes de la preuve de Goguen, nous renvoyons le lecteur à l'annexe A, et pour voir la preuve complète, nous le renvoyons à [Gog94].

3.2 la sémantique opérationnelle : le système UTT^{rS}

Nous introduisons ici le système UTT^{rS} qui représente une sémantique opérationnelle typée pour le calcul UTT^r.

3.2.1 Définitions pour la métathéorie

Dans un premier temps, nous présentons plusieurs notions métathéoriques qui seront importantes pour la présentation de UTT^{rS} ainsi que pour exposer les propriétés de UTT^r.

Définition 3.1 (Sous-dérivation)**.** *Soient deux dérivations J_1 et J_2. J_1 est une sous-dérivation de J_2 si J_1 apparaît comme un sous-arbre de J_2.*

Définition 3.2 (Equivalence syntaxique)**.** *Deux termes sont équivalents syntaxiquement, $M \equiv N$, s'ils sont identiques au renommage des variables liées près.*

Définition 3.3 (Famille de Pré-schémas)**.** *Soit X une variable.*

- *Une sorte Φ est un opérateur pré-positif par rapport à X,* notation $PPOS_X(\Phi)$, *si :*
  - $\Phi \equiv El(X)$ *ou*
  - $(x : A)\Phi_0$*, où A est une petite sorte et* $PPOS_X(\Phi_0)$.
- *Une sorte Θ est un pré-schéma inductif par rapport à X,* notation $PSCH_X(\Theta)$, *si :*
  - $\Theta \equiv El(X)$,
  - $(x : A)\Theta_0$*, où A est une petite sorte et* $PSCH_X(\Theta_0)$*, ou*
  - $(\Phi)\Theta_0$*, où* $PPOS_X(\Phi)$ *et* $PSCH_X(\Theta_0)$.
- *Une séquence finie de sortes $\langle\Theta_1, \dots, \Theta_n\rangle$, notée $\bar{\Theta}$, est une famille de pré-schémas par rapport à X,* notation $PSCH_X(\Theta_1, \dots, \Theta_n)$*, si* $PSCH_X(\Theta_i)$ *pour* $1 \leq i \leq n$.

Définition 3.4 (Réductions non typées)**.** *On introduit les relations de récriture suivantes :*

- $([x : A_1]M_0)(M_2) \quad \beta \quad [M_2/x]M_0$
- $[x : A_1]M(x) \quad \eta \quad M \qquad x \notin FV(M)$
- $E_\forall(A_1, P_1, R, f, \Lambda(A_2, P_2, g)) \quad o \quad f(g)$
- $E^X[\bar{\Theta}_1](C, \bar{f}, \iota_i^X[\bar{\Theta}_2](\bar{a})) \quad o \quad f_i(\bar{a}, \Phi_{i_1}^{\natural}[M^X[\bar{\Theta}_3], C, E^X[\bar{\Theta}_3](C', \bar{f}'), a_i], \dots)$
- $T(prop) \quad o \quad Prop$
- $T(prf(P)) \quad o \quad Prf(P)$

- $T(\mu^X[\bar{\Theta}]) \quad o \quad M^X[\bar{\Theta}]$
- $E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)) \quad \vartheta$
  $E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k)$

On note $\rhd$ pour désigner la fermeture contextuelle de l'ensemble des récritures précédentes. Un terme M est un rédex s'il existe N tel que $M \beta\eta\vartheta o N$.

Définition 3.5. *Soit R une relation de récriture sur les termes, la réduction $\rhd_R$ est sa fermeture contextuelle.*
On note $M \rhd^+_R N$ pour la fermeture transitive de la réduction $\rhd_R$ et $M \rhd^*_R N$ pour la fermeture réflexive transitive de $\rhd_R$.

Définition 3.6. *Un terme est fortement normalisant si toute séquence de réductions partant de ce terme termine.*

Définition 3.7 (Pre-redex)**.** *On dit qu'un terme M est un full pre-redex si M est une abstraction ou de la forme $E_\forall(A, P, R, f)$, $E^X[\bar{\Theta}](C, \bar{f})$ ou T.*
Si M est un full pre-redex et que M n'est pas une abstraction alors on dit que M est un object-level full pre-redex.
Un terme qui est un full pre-redex ou un sous-terme de $E_\forall(A, P, R, f)$ ou $E^X[\bar{\Theta}](C, \bar{f})$ est un pre-redex.

Définition 3.8 (Terme de base)**.** *On définit la notion de terme de base inductivement de la manière suivante :*

- *Les variables sont des termes de base.*
- *Si M_1 est un terme de base alors $M_1(M_2)$ est un terme de base.*
- *Si M_2 est un terme de base et M_1 est un object-level full pre-redex alors $M_1(M_2)$ est un terme de base.*

Définition 3.9 (Forme normale de tête faible)**.** *Un terme M est en forme normale de tête faible si M est un terme de base ou un pre-redex.*

Définition 3.10 (Forme normale)**.** *Un terme M ou une sorte A est en forme normale si :*

- *M est une variable.*
- *$A \equiv (x : A_1)A_2$, où A_1 et A_2 sont en forme normale ;*
- *$M \equiv [x : A_1]M_0$, où A_1 est normale et M_0 est normal et pas de la forme $N(x)$ avec $x \notin FV(N)$;*
- *$M \equiv M_1(M_2)$, où M_1 et M_2 sont en forme normale et $M_1(M_2)$ n'est pas un redex ;*
- *$A \equiv$ **Type** ;*
- *$A \equiv El(M)$, et M est normal ;*
- *M est une constante ;*
- *$M \equiv \kappa^X[\bar{\Theta}]$, où Θ_i est normale pour $1 \leq i \leq n$.*

Lemme 3.11. *Si M est en forme normale alors M n'admet aucune réduction.*

Démonstration. Par induction sur les termes en forme normale.

□

A cause de la présence des règles de substitution qui sont admissibles, le système UTT^r n'est pas le plus pratique pour raisonner. On introduit donc le calcul UTT^{r-} dont les jugements sont les mêmes que ceux de UTT^r mais qui ne contient pas de règles de substitution. Le symbole $\vdash^-$ est employé pour indiquer qu'un jugement a été dérivé dans UTT^-. Il faut donc prouver la complétude de UTT^r par rapport à UTT^{r-}.

Lemme 3.12 (Complétude de UTT^r par rapport à UTT^{r-})**.** *Si le jugement $\Gamma \vdash^- J$ est dérivable alors $\Gamma \vdash J$ est dérivable.*

Démonstration. Par induction sur les dérivations, à l'aide du lemme 4.23 du chapitre 4.

□

3.2.2 Jugements et dérivations

Les formes de jugement pour UTT^{rS} et leurs significations intuitives sont les suivantes :

- $\Gamma \vdash^S$ **valid**, signifie que Γ est un contexte valide ;
- $\Gamma \vdash^S A \xrightarrow{nf} B$, signifie que la sorte A admet une forme normale qui est la sorte B et que A et B sont des sortes valides dans le contexte Γ ;
- $\Gamma \vdash^S M \xrightarrow{nf} P : A$, signifie que le terme M admet une forme normale P de sorte A dans le contexte Γ ;
- $\Gamma \vdash^S M \xrightarrow{wh} N : A$, signifie que M se réduit en N en un pas, par une stratégie externe de réduction, de plus M et N sont des termes de sorte A dans le contexte Γ.

Nous considérons également les abréviations suivantes :

- $\Gamma \vdash^S M : B$, s'il existe un terme P tel que $\Gamma \vdash^S M \xrightarrow{nf} P : B$;
- $\Gamma \vdash^S A$**kind**, s'il existe une sorte B telle que $\Gamma \vdash^S A \xrightarrow{nf} B$;
- $\Gamma \vdash^S M \downarrow N[P] : B$, si $\Gamma \vdash^S M \xrightarrow{nf} P : B$ et $\Gamma \vdash^S N \xrightarrow{nf} P : B$;
- $\Gamma \vdash^S A \downarrow B[C]$, si $\Gamma \vdash^S A \xrightarrow{nf} C$ et $\Gamma \vdash^S B \xrightarrow{nf} C$;
- $SCH^S_{\Gamma,X}(\bar{\Theta})$, si $\Gamma \vdash^S$ **valid**, $PSCH_X(\bar{\Theta})$ et $\Gamma, X :$ **Type** $\vdash^S \Theta_i \xrightarrow{nf} \Theta'_i$ pour $1 \leq i \leq n$;
- $\Gamma \vdash^S \bar{\Theta} \downarrow \bar{\Theta}'[\bar{\Theta}'']$, si $\Gamma \vdash^S$ **valid**, $PSCH_X(\bar{\Theta})$, $PSCH_X(\bar{\Theta}')$ et $\Gamma, X :$ **Type** $\vdash^S \Theta_i \downarrow \Theta'_i[\bar{\Theta}'']$ pour $1 \leq i \leq n$.

UTT^{rS} est définie comme la plus petite relation définie par les règles d'inférence qui vont suivre :

- **Contexte**

$$\frac{}{<>\vdash^S \textbf{valid}}\ \text{(S-Emp)} \qquad \frac{\Gamma \vdash^S A \xrightarrow{nf} B \qquad x \notin FV(\Gamma)}{\Gamma, x : A \vdash^S \textbf{valid}}\ \text{(S-Weak)}$$

- **Forme canonique**

$$\frac{\Gamma, x : A, \Gamma' \vdash^S A \xrightarrow{nf} B}{\Gamma, x : K, \Gamma' \vdash^S x \xrightarrow{nf} x : B}\ \text{(S-Var)}$$

$$\frac{\Gamma \vdash^S A_1 \xrightarrow{nf} B_1 \qquad \Gamma, x : A_1 \vdash^S A_2 \xrightarrow{nf} B_2}{\Gamma \vdash^S (x : A_1)A_2 \xrightarrow{nf} (x : B_1)B_2}\ \text{(S-}\Pi\text{)}$$

$$\frac{\Gamma \vdash^S A_1 \xrightarrow{nf} B_1 \qquad \Gamma, x : A_1 \vdash^S M_0 \xrightarrow{nf} N_0 : B_2}{\Gamma \vdash^S [x : A_1]M_0 \xrightarrow{nf} [x : B_1]N_0 : (x : B_1)B_2}\ \text{(S-}\lambda\text{)}$$

avec $N_0 \not\equiv N(x)$ avec $x \notin FV(N)$.

$$\frac{\Gamma \vdash^S A_1 \xrightarrow{nf} B_1 \qquad \Gamma, x : A_1 \vdash^S M_0 \xrightarrow{nf} N(x) : B_2 \qquad \Gamma \vdash^S N \xrightarrow{nf} P : (x : B_1)B_2}{\Gamma \vdash^S [x : A_1]M_0 \xrightarrow{nf} P : (x : B_1)B_2}\ \text{(S-}\eta\text{)}$$

$$\frac{\Gamma \vdash^S M_1 \xrightarrow{nf} N_1 : (x : B_1)B_2 \qquad \Gamma \vdash^S M_2 \xrightarrow{nf} N_2 : B_1 \qquad \Gamma \vdash^S [M_2/x]B_2 \xrightarrow{nf} C}{\Gamma \vdash^S M_1(M_2) \xrightarrow{nf} N_1(N_2) : C}\ \text{(S-App)}$$

où $M_1(M_2)$ est en forme normale de tête faible.

$$\frac{\Gamma \vdash^S \textbf{valid}}{\Gamma \vdash^S \textbf{Type} \xrightarrow{nf} \textbf{Type}}\ \text{(S-Type)} \qquad \frac{\Gamma \vdash^S M \xrightarrow{nf} N : \textbf{Type}}{\Gamma \vdash^S El(M) \xrightarrow{nf} El(N)}\ \text{(S-El)}$$

$$\frac{\Gamma \vdash^S \textbf{valid}}{\Gamma \vdash^S c \xrightarrow{nf} c : A}\ \text{(S-C)}$$ où c est une constante de sorte A.

$$\frac{\Gamma \vdash^S M \xrightarrow{wh} N : B \qquad \Gamma \vdash^S N \xrightarrow{nf} P : B}{\Gamma \vdash^S M \xrightarrow{nf} P : B} \text{ (S-WH)}$$

$$\frac{\Gamma \vdash^S A \xrightarrow{nf} B \quad x \notin FV(\Gamma)}{\Gamma, x : A \vdash^S \textbf{valid}} \text{ (S-Weak)}$$

- **Forme de schéma canonique**

$$\frac{\Gamma \vdash^S SCH^S_{\Gamma,X}(\bar{\Theta})}{\Gamma \vdash^S M^X[\bar{\Theta}] \xrightarrow{nf} M^X[\bar{\Theta}']} \text{ (S-M)}$$

$$\frac{\Gamma \vdash^S SCH^S_{\Gamma,X}(\bar{\Theta})}{\Gamma \vdash^S \iota^X_i[\bar{\Theta}] \xrightarrow{nf} \iota^X_i[\bar{\Theta}'] : \Theta(M^X[\bar{\Theta}'])} \text{ (S-i)} \quad \text{avec } 1 \leq i \leq n$$

$$\frac{\Gamma \vdash^S SCH^S_{\Gamma,X}(\bar{\Theta})}{\Gamma \vdash^S E^X[\bar{\Theta}] \xrightarrow{wh} E^X[\bar{\Theta}'] : (C : (M^X[\bar{\Theta}'])\textbf{Type})} \text{ (S-E)}$$

$$(f_1 : (\Theta'_1)^\circ[M^X[\bar{\Theta}'], C, i^X_1[\bar{\Theta}']])$$

$$\ldots$$

$$(f_n : (\Theta'_n)^\circ[M^X[\bar{\Theta}'], C, i^X_n[\bar{\Theta}']])(z : M^X[\bar{\Theta}'])C(z)$$

$$\frac{\Gamma \vdash^S \bar{\Theta} \downarrow \bar{\Theta}'[\bar{\Theta}'']}{\Gamma \vdash^S \mu^X[\bar{\Theta}] \xrightarrow{nf} \mu^X[\bar{\Theta}''] : \textbf{Type}} \text{ (S-}\mu\text{)}$$

où $(\Gamma', A) \in TYPES_\Gamma(\bar{\Theta}')$ implique $A \equiv T(\alpha)$.

- **Réduction de tête faible**

$$\frac{\Gamma \vdash^S [x : A_1]M_0 : (x : B_1)B_2 \qquad \Gamma \vdash^S M_2 : B_1 \qquad \Gamma \vdash^S [M_2/x]B_2 \xrightarrow{nf} C}{\Gamma \vdash^S ([x : A_1]M_0)(M_2) \xrightarrow{wh} [M_2/x]M_0 : C} \text{ (W-}\beta\text{)}$$

$$\frac{\begin{array}{c}\Gamma \vdash^S E_\forall(A_1, P_1, R, f) \xrightarrow{nf} E_\forall(A_3, P_3, R', f')) : (M : Prf(\forall(A_3, P_3)))S \\ \Gamma \vdash^S \Lambda(A_2, P_2, g) \xrightarrow{nf} \Lambda(A_3, P_3, g') : Prf(\forall(A_3, P_3, g')) \\ \Gamma \vdash^S Prf(R(\Lambda(A_3, P_3, g'))) \xrightarrow{nf} B\end{array}}{\Gamma \vdash^S E_\forall(A_1, P_1, R, f, \Lambda(A_2, P_2, g)) \xrightarrow{wh} f(g) : B} \text{(W-}E_\forall\text{)}$$

$$\frac{\begin{array}{c}\Gamma \vdash^S E^X[\bar{\Theta}_1](C, \bar{f}) \xrightarrow{nf} E^X[\bar{\Theta}_3](C', \bar{f}') : (M : M^X[\bar{\Theta}_3])D \\ \Gamma \vdash^S \iota_i^X[\bar{\Theta}_2](\bar{a}) \xrightarrow{nf} \iota_i^X[\bar{\Theta}_3](\bar{a'}) : M^X[\bar{\Theta}_3] \\ \Gamma \vdash^S C(\iota_i^X[\bar{\Theta}_2](\bar{a})) \xrightarrow{nf} B\end{array}}{\Gamma \vdash^S E^X[\bar{\Theta}_1](C, \bar{f}, \iota_i^X[\bar{\Theta}_2](\bar{a})) \xrightarrow{wh} f_i(\bar{a}, \Phi^\natural_{i_1}[M^X[\bar{\Theta}_3], C, E^X[\bar{\Theta}_3](C', \bar{f}'), a_i], \dots) : B} \text{(W-}E^X[\bar{\Theta}]\text{)}$$

$$\frac{\Gamma \vdash^S prop : El(U)}{\Gamma \vdash^S T(prop) \xrightarrow{wh} Prop : \textbf{Type}} \text{(W-prop)}$$

$$\frac{\Gamma \vdash^S prf(P) : El(U)}{\Gamma \vdash^S T(prf(P)) \xrightarrow{wh} Prf(P) : \textbf{Type}} \text{(W-prf)}$$

$$\frac{\Gamma \vdash^S \mu^X[\bar{\Theta}] : El(U)}{\Gamma \vdash^S T(\mu^X[\bar{\Theta}]) \xrightarrow{wh} M^X[\bar{\Theta}] : \textbf{Type}} \text{(W-}\mu\text{)}$$

$$\frac{\Gamma \vdash^S M_1 \xrightarrow{wh} N_1 : (x : A_1)A_2 \qquad \Gamma \vdash^S M_2 : A_1 \qquad \Gamma \vdash^S [M_2/x]A_2 \xrightarrow{nf} B}{\Gamma \vdash^S M_1(M_2) \xrightarrow{wh} N_1(M_2) : B} \text{(W-App)}$$

$$\frac{\Gamma \vdash^S M_1 : (x : B_1)B_2 \qquad \Gamma \vdash^S M_2 \xrightarrow{wh} N_2 : B_1 \qquad \Gamma \vdash^S [M_2/x]B_2 \xrightarrow{nf} C}{\Gamma \vdash^S M_1(M_2) \xrightarrow{wh} M_1(N_2) : C} \text{(W-Obj)}$$

où M_1 est un full object-level pre-redex.

- **Nouvelle réduction**

$$\frac{\begin{array}{c}\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m) \xrightarrow{nf} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \\ \Gamma \vdash^S E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) \xrightarrow{nf} E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m \\ \Gamma \vdash^S k \xrightarrow{nf} v : F_n \quad v \in FV(\Gamma)\end{array}}{\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)) \xrightarrow{wh} E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A} \text{(W-}\vartheta\text{)}$$

Avec :
$\bar{\Theta}_m \equiv < \underbrace{X, \ldots, X}_{m fois} >$, $\bar{\Theta}_n \equiv < \underbrace{X, \ldots, X}_{n fois} >$,
$\sigma : [n] \rightarrow [m]$,
$C \equiv [y : F_m]A : (y : F_m)$**Type**,
$C_1 \equiv [z : F_n]F_m : (z : F_n)$**Type**,
$C_2 \equiv [x : F_n]A : (x : F_n)$**Type**, et tous les a_i sont en forme normale.

3.2.3 Propriétés de UTT^{rS}

Nous présentons ici plusieurs propriétés de UTT^{rS}.
Le lemme 3.13(Génération) est très important dans l'établissement de la sémantique opérationnelle typée UTT^{rS}, car il intervient dans la preuve de nombreux lemmes. Nous avons veillé lors de l'ajout de notre nouvelle règle à préserver cette propriété.

Lemme 3.13 (Génération)**.** *Si un jugement est dérivable dans UTT^{rS}, alors quelque soit la dérivation de ce jugement, elle se termine par une dernière règle d'inférence unique et déterminée .*

Démonstration. Preuve par induction sur les dérivations.

Nous partons du principe que cette preuve est déjà réalisée dans le cadre du système UTT^S.
Prenons un jugement qui peut être généré par la nouvelle règle (W-ϑ) et voyons si une autre règle pourrait générer ce même jugement.
La règle (W-ϑ) représente une réduction concernant des types inductifs faisant intervenir leurs termes de recursion. Naturellement la seule règle avec laquelle elle pourrait être en concurrence serait la règle (W-$E^X[\bar{\Theta}]$).
Si le jugement

$$\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) \xrightarrow{wh} E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A$$

est dérivable par application de la règle (W-ϑ) , on a forcément pour prémisse $\Gamma \vdash^S k \xrightarrow{nf} v : F_n \quad v \in FV(\Gamma)$. Premièrement k ne peut pas être de la forme $\iota^X_i[\bar{\Theta}_n]$, par conséquent $E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)$ n'est pas de la forme $\iota^X_i[\bar{\Theta}_m]$. On ne peut donc pas employer la règle (W-$E^X[\bar{\Theta}]$) pour l'application du terme $E[\bar{\Theta}_m](C, a_1, ..., a_m)$ au terme $E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)$.
Deuxièmement, la forme normale de k étant une variable v, $E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)$ ne peut se réduire en un terme de la forme canonique $\iota^X_i[\bar{\Theta}_m]$. Il ne sera pas non plus possible d'appliquer la règle (W-$E^X[\bar{\Theta}]$) après réduction du terme $E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)$ dans l'expression $E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)$ par la règle (W-Obj). □

Corollaire 3.14. *Si un jugement est dérivable dans UTT^{rS}, alors il existe une et une seule dérivation de ce jugement dans UTT^{rS}.*

Lemme 3.15 (Unicité de la forme normale)**.**

- *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : B$ *et* $\Gamma \vdash^S M \xrightarrow{nf} Q : C$ *alors* $P \equiv Q$ *et* $B \equiv C$.
- *Si* $\Gamma \vdash^S A \xrightarrow{nf} B$ *et* $\Gamma \vdash^S A \xrightarrow{nf} C$ *alors* $B \equiv C$.

Démonstration. Par induction sur les dérivations utilisant le lemme 3.13(Génération). □

Lemme 3.16 (Validité de contexte)**.** *Toute dérivation du jugement* $\Gamma_0, \Gamma_1 \vdash^S J$ *admet une sous-dérivation du jugement* $\Gamma_0 \vdash^S$ ***valid****.*

Démonstration. Preuve par induction sur les dérivations.

$\square$

Corollaire 3.17. *Si le jugement $\Gamma_0, x : A, \Gamma_1 \vdash^S J$ est dérivable dans UTT^{rS}, alors il existe une sorte B telle qu'il existe une sous-dérivation du jugement $\Gamma_0 \vdash^S A \xrightarrow{nf} B$.*

Démonstration. Par le lemme 3.16, on sait qu'il existe une dérivation de $\Gamma_0, x : A \vdash^S$ **valid**. De plus, grâce au lemme 3.13, on sait qu'il existe une sous-dérivation de $\Gamma_0 \vdash^S A \xrightarrow{nf} B$ pour un certain B, par rapport à la règle (S-Weak).

$\square$

Lemme 3.18 (Complétude pour UTT^{r-})**.**

- *Si $\Gamma \vdash^S$ **valid** alors $\Gamma \vdash^-$ **valid**.*
- *Si $\Gamma \vdash^S A \xrightarrow{nf} B$ alors $\Gamma \vdash^- A$**kind** et $\Gamma \vdash^- A = B$.*
- *Si $\Gamma \vdash^S M \xrightarrow{nf} P : A$ alors $\Gamma \vdash^- M : A$ et $\Gamma \vdash^- M = P : A$.*
- *Si $\Gamma \vdash^S M \xrightarrow{wh} N : A$ alors $\Gamma \vdash^- M : A$ et $\Gamma \vdash^- M = N : A$.*

Démonstration. Preuve par induction sur les dérivations.
Considérons le cas de la règle (W-ϑ). Par hypothèse d'induction nous avons les jugements suivants :

- (1) $\Gamma \vdash^- E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A$,
- (2) $\Gamma \vdash^- E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m$,
- (3) $\Gamma \vdash^- k : F_n$,
- (4) $\Gamma \vdash^- E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) : F_m$

Dans un premier temps, les jugements (1) et (4) nous permettent de dériver le jugement $\Gamma \vdash^- E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) : A$ en appliquant la règle (5.5). Dans un second temps, à partir des jugements (1),(2) et (3), en appliquant la règle (ϑ-eq), nous pouvons dériver

$\Gamma \vdash^{-} E[\bar{\Theta}_m]([x : F_m]A, a_1, ..., a_n)(E[\bar{\Theta}_n]([x : F_n]F_m, b_{\sigma(1)}, ..., b_{\sigma(n)})(k) = E[\bar{\Theta}_n]([x : F_n]A, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A$.

□

Lemme 3.19 (Contexte)**.** *Si le jugement $\Gamma \vdash^S J$ est dérivable, alors Γ est un contexte. De plus :*

- *si $\Gamma \vdash^S A \xrightarrow{nf} B$, alors $FV(A) \subseteq dom(\Gamma)$ et $FV(B) \subseteq dom(\Gamma)$;*
- *si $\Gamma \vdash^S M \xrightarrow{nf} P : B$, alors $FV(M) \subseteq dom(\Gamma)$, $FV(P) \subseteq dom(\Gamma)$ et $FV(B) \subseteq dom(\Gamma)$;*
- *si $\Gamma \vdash^S M \xrightarrow{wh} N : B$, alors $FV(M) \subseteq dom(\Gamma)$, $FV(N) \subseteq dom(\Gamma)$ et $FV(B) \subseteq dom(\Gamma)$.*

Démonstration. Preuve par induction sur les dérivations.

□

Définition 3.20 (Renommage)**.** *Un renommage est une substitution δ de Δ vers Γ telle que pour chaque $(x : A) \in \Gamma$ nous avons $\delta(x) \equiv y$ et $(y : \hat{\delta}(A)) \in \Delta$.*
$\hat{\delta}(M)$ représente M dans lequel toutes les variables libres v sont remplacées par $\delta(v)$.

Lemme 3.21.

- *Si δ est un renommage de Δ vers Γ et ϕ est un renommage de Φ vers Δ alors $\phi \circ \delta$ est un renommage de Φ vers Γ.*
- *Si M est un terme en forme normale de tête faible et δ est un renommage de Δ vers Γ alors $\hat{\delta}(M)$ est normal de tête faible.*

Démonstration. Voir [Gog94].

□

Lemme 3.22 (Renommage)**.** *Si δ est un renommage de Δ vers Γ alors :*

- *si $\Gamma \vdash^S A \xrightarrow{nf} B$, alors $\Delta \vdash^S \hat{\delta}(A) \xrightarrow{nf} \hat{\delta}(B)$;*

- *si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$*, alors* $\Delta \vdash^S \hat{\delta}(M) \xrightarrow{nf} \hat{\delta}(P) : \hat{\delta}(A)$.

Démonstration. Preuve par induction sur les dérivations.

□

Corollaire 3.23. *Si le jugement* $\Gamma \vdash^S J$ *est dérivable, et que* Δ *est un contexte valide qui contient tous les composants de* Γ*, alors le jugement* $\Delta \vdash^S J$ *est dérivable.*

Démonstration. Par induction sur les dérivations.

□

Lemme 3.24 (Remplacement de contexte)**.** *Si* $\Gamma_0 \vdash^S A \downarrow B[C]$ *et* $\Gamma_0, x : A, \Gamma_1 \vdash^S J$*, alors* $\Gamma_0, x : B, \Gamma_1 \vdash^S J$.

Démonstration. Par induction sur les dérivations.

□

Lemme 3.25 (Renforcement)**.** *On suppose que* z *est une variable telle que* $z \notin FV(\Gamma_1)$*, alors :*

- *si* $\Gamma_0, z : C, \Gamma_1 \vdash^S$ ***valid****, alors* $\Gamma_0, \Gamma_1 \vdash^S$ ***valid*** *;*

- *si* $\Gamma_0, z : C, \Gamma_1 \vdash^S A \xrightarrow{nf} B$ *et* $z \notin FV(A)$*, alors* $\Gamma_0, \Gamma_1 \vdash^S A \xrightarrow{nf} B$ *;*

- *si* $\Gamma_0, z : C, \Gamma_1 \vdash^S M \xrightarrow{nf} P : A$ *et* $z \notin FV(M)$*, alors* $\Gamma_0, \Gamma_1 \vdash^S M \xrightarrow{nf} P : A$*;*

- *si* $\Gamma_0, z : C, \Gamma_1 \vdash^S M \xrightarrow{wh} N : A$ *et* $z \notin FV(M)$*, alors* $\Gamma_0, \Gamma_1 \vdash^S M \xrightarrow{wh} N : A$.

Démonstration. Par induction sur les dérivations.

□

Le calcul UTT^{rS} étant une synthèse des réductions et des règles de typage de UTT^{rS}, nous pouvons montrer la relation entre les réductions non typées et le typage dans UTT^{rS}.

Lemme 3.26 (Adequation pour les réductions non typées)**.**

- *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$*, alors* P *est en forme normale et il existe un* N *tel que* $M \rhd^*_{\beta o \vartheta} N \rhd^*_{\eta} P$.
- *Si* $\Gamma \vdash^S A \xrightarrow{nf} C$*, alors* C *est en forme normale et il existe un* B *tel que* $A \rhd^*_{\beta o \vartheta} B \rhd^*_{\eta} C$.
- *Si* $\Gamma \vdash^S M \xrightarrow{wh} N : A$*, alors* $M \rhd_{\beta o \vartheta} N$.

Démonstration. Par induction sur les dérivations.

□

Corollaire 3.27.

- *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$ *et* M *est en forme normale, alors* $M \equiv P$.
- *Si* $\Gamma \vdash^S A \xrightarrow{nf} B$ *et* A *est en forme normale, alors* $A \equiv B$.

Démonstration. Considérons le premier cas. Grâce au lemme 3.26 on sait que $M \rhd^* P$. De plus M étant en forme normale, M n'admet aucune réduction (lemme 3.11).

□

Nous présentons ici deux prédicats utilisés par Goguen dans sa preuve. Ces prédicats dans leur définition incluent les notions de normalisation forte, de préservation de type et de confluence.

Définition 3.28. *Les prédicats* $S^C_\Gamma(A)$ *et* $S^{P,A}_\Gamma(M)$ *sont définis comme étant la plus petite relation telle que :*

- *on a* $S^C_\Gamma(A)$*, si* $\Gamma \vdash^S A \xrightarrow{nf} C$ *et pour tout* B *si* $A \rhd B$ *alors* $S^C_\Gamma(B)$ *;*
- *on a* $S^{P,A}_\Gamma(M)$ *si*
  - $\Gamma \vdash^S M \xrightarrow{nf} P : A$,

– *pour tout N si $M \rhd N$ alors $S_\Gamma^{P,A}(N)$, et*

– *pour tout M_i sous terme de M, il existe B et Q tels que $\Gamma, \Delta \vdash^S M_i \xrightarrow{nf} B : Q$.*

$C_\Gamma^C(A)$ et $C_\Gamma^{P,A}(M)$ sont les mêmes prédicats que $S_\Gamma^C(A)$ et $S_\Gamma^{P,A}(M)$ à la différence que, au lieu de $A \rhd B$ et $M \rhd N$, nous avons $A \rhd^+ B$ et $M \rhd^+ N$.

Lemme 3.29 (Propriétés des prédicats)**.**

- *Si $S_\Gamma^{P,A}(M)$, alors $\Gamma \vdash^S M \xrightarrow{nf} P : A$.*
- *Si $S_\Gamma^{P,A}(M)$, alors M est fortement normalisant.*
- *Si $S_\Gamma^{P,A}(M)$ et $M \rhd^* N$, alors $S_\Gamma^{P,A}(N)$.*
- *On a $S_\Gamma^{P,A}(M)$ si et seulement si $C_\Gamma^{P,A}(M)$.*
- *Si $S_\Gamma^{P,A}(M)$ et N est un sous-terme de M, alors il existe Q et B tels que $S_\Gamma^{Q,B}(N)$.*
- *Si $\Gamma_0 \vdash^S C \downarrow D[E]$ alors :*
  – *si $S^B_{\Gamma_0,x:C,\Gamma_1}(A)$, alors $S^B_{\Gamma_0,x:D,\Gamma_1}(A)$ et*
  – *si $S^{P,A}_{\Gamma_0,x:C,\Gamma_1}(M)$, alors $S^{P,A}_{\Gamma_0,x:D,\Gamma_1}(M)$.*
- *Si z est une variable telle que $z \notin FV(\Gamma_1)$ alors :*
  – *si $S^B_{\Gamma_0,x:C,\Gamma_1}(A)$ et $z \notin FV(A)$, alors $S^B_{\Gamma_0,\Gamma_1}(A)$ et*
  – *si $S^{P,A}_{\Gamma_0,x:C,\Gamma_1}(M)$ et $z \notin FV(M)$, alors $S^{P,A}_{\Gamma_0,\Gamma_1}(M)$.*

Démonstration. Par induction sur les preuves de $S_\Gamma^C(A)$ et $S_\Gamma^{P,A}(M)$. □

Remarque 3.30. *Le résultat le plus important de cette section est le lemme 3.32. C'est en effet ce lemme qui établit le fait que tous les termes et les sortes de UTT^{rS}*

vérifient les deux prédicats de la définition 3.28, prédicats, qui nous le rappelons, incluent toutes les propriétés désirées dans leur définition. La preuve faite par Goguen dans sa thèse pour ce lemme dans le cas du système UTT^S, manque selon nous, de clarté notamment au niveau du principe d'induction qu'il utilise. Nous avons donc complètement revu la structure de cette preuve. Afin d'améliorer la preuve du lemme 3.32 que nous devons de toute façon modifier - car l'ajout de notre nouvelle reduction nous y oblige - nous avons jugé nécessaire l'ajout du 3.31. De plus, il nous a fallu proposer un nouveau principe d'induction que l'on présente dans la remarque 3.33.

Lemme 3.31 (Jugements sous-termes). *Si $\Gamma \vdash^S M \xrightarrow{nf} P : A$, $\Gamma \vdash^S M \xrightarrow{wh} P : A$ ou $\Gamma \vdash^S M \xrightarrow{nf} P$, alors pour tout M_i sous-terme de M il existe B et Q tels que $\Gamma, \Delta \vdash^S M_i \xrightarrow{nf} B : Q$ ou $\Gamma, \Delta \vdash^S M_i \xrightarrow{nf} B$.*

Démonstration. Par induction sur les dérivations. On parcourt les formes possibles de M.

- Si $M \equiv X$, où X est une variable, la conclusion est évidente.
- Si $M \equiv [x : A]M_1$, le jugement $\Gamma \vdash^S [x : A]M \xrightarrow{nf} P$ est obtenu par une des règles suivantes :
  - (S-λ), dans ce cas on retrouve comme prémisses $\Gamma \vdash^S A \xrightarrow{nf} B$ et $\Gamma, x : A \vdash^S M \xrightarrow{nf} N : B$;
  - (S-η), dans ce cas on retrouve comme prémisses $\Gamma \vdash^S A \xrightarrow{nf} B$ et $\Gamma, x : A \vdash^S M \xrightarrow{nf} N(x) : B_2$.
- Si $M \equiv M_1(M_2)$, le jugement $\Gamma \vdash^S M_1(M_2) \xrightarrow{nf/wh} P : A$ est obtenu par une des règles suivantes :
  - (S-App), dans ce cas on retrouve comme prémisses $\Gamma \vdash^S M_1 \xrightarrow{nf} N_1 : (x : B_1)B_2$ et $\Gamma \vdash^S M_2 \xrightarrow{nf} N_2 : B_1$;
  - (W-β), dans ce cas $M_1(M_2)$ est de la forme $([x : A]N_1)(N_2)$, et on retrouve comme prémisses $\Gamma \vdash^S [x : A]N_1 : (x : B_1)B_2$ et $\Gamma \vdash^S N_2 : B_1$;

- (W-E[Θ]), ici $M_1(M_2)$ est de la forme $E[\Theta](c,\overline{f})(\iota_i^X[\Theta](\overline{a}))$, on retrouve comme prémisses $\Gamma \vdash^S E[\Theta](c,\overline{f}) \xrightarrow{nf} P : A$ et $\Gamma \vdash^S (\iota_i^X[\Theta](\overline{a})) \xrightarrow{nf} Q : B$;

- (W-prop), ici $M_1(M_2)$ est de la forme $T(prop)$, où T est une constante et comme prémisse on a $\Gamma \vdash^S prop : El(U)$;

- (W-prf), ici $M_1(M_2)$ est de la forme $T(prf(P))$, où T est une constante et comme prémisse on a $\Gamma \vdash^S prf(P) : El(U)$;

- (W-μ), ici $M_1(M_2)$ est de la forme $T(\mu[\Theta])$, où T est une constante et comme prémisse on a $\Gamma \vdash^S \mu[\Theta] : El(U)$;

- (W-App), on retrouve comme prémisses $\Gamma \vdash^S M_1 \xrightarrow{wh} N_1 : (x : A_1)A_2$ et $\Gamma \vdash^S M_2 \xrightarrow{nf} N_2 : A_1$;

- (W-obj), on retrouve comme prémisses $\Gamma \vdash^S M_1 \xrightarrow{wh} N_1 : (x : A_1)A_2$ et $\Gamma \vdash^S M_2 \xrightarrow{wh} N_2 : A_1$;

- (W-ϑ), ici $M_1(M_2)$ est de la forme $E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)$ et on retrouve comme prémisses $\Gamma \vdash^S E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m$, $\Gamma \vdash^S k \xrightarrow{nf} v : F_n$ et $\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A$.

- Si M est de la forme $M^X[\Theta]$,$\iota_i^X[\Theta]$,$E[\Theta]$ ou $\mu[\Theta]$ qui sont des expressions constantes de la forme $\kappa[\Theta]$, alors les jugements $\Gamma \vdash^S \kappa[\Theta] \xrightarrow{nf} P : A$ sont dérivés par des règles dont la prémisse est $SCH^S_{\Gamma,X}(\overline{\Theta})$. Par définition de $SCH()$ et par hypothèse d'induction, on peut conclure.

- Si $M \equiv El(M_1)$, on dérive le jugement $\Gamma \vdash^S El(M_1) \xrightarrow{nf} El(N)$ par la règle (S-El) qui a pour prémisse $\Gamma \vdash^S M_1 \xrightarrow{nf} N$: **Type**.

- Si $M \equiv (x : El(M_1))El(M_2)$, on dérive le jugement $\Gamma \vdash^S (x : El(M_1))El(M_2) \xrightarrow{nf} (x : El(M_1'))El(M_2')$ par la règle (S-Π) qui a pour prémisses $\Gamma \vdash El(M_1) \xrightarrow{nf}$

$El(M_1')$ et $\Gamma, x : El(M_1) \vdash El(M_2) \xrightarrow{nf} El(M_2')$, par hypothèse d'induction on peut conclure.

- Si M est de la forme T, prf, $prop$, $Prop$ ou **Type** qui sont des constantes, la règle (S-C) nous permet de conclure.

□

Lemme 3.32 (Réduction)**.**

- *Si* $\Gamma \vdash^S A \xrightarrow{nf} B$ *alors* $S_\Gamma^B(A)$.
- *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$ *alors* $S_\Gamma^{P,A}(M)$.
- *Si* $\Gamma \vdash^S M \xrightarrow{wh} N : A$ *et* $S_\Gamma^{P,A}(N)$ *alors* $S_\Gamma^{P,A}(M)$.

Remarque 3.33 (Nouvelle induction)**.** *Dans la preuve qui suit nous serons amenés à utiliser l'induction suivante.*
Considérons l'ensemble des n-uplets $(A_1, \ldots, A_n)$*, où les* A_i *(pour* $1 \leq i \leq n$*), sont des arbres, muni de la relation d'ordre* $\prec$*. On a* $(A_1, \ldots, A_n) \prec (A_1', \ldots, A_n')$ *si et seulement si, il existe un* i *tel que pour tout* $j < i$ *on a* $A_j = A_j' \wedge A_i < A_i'$*. L'expression* $(A_i < A_i')$ *signifie que* A_i *est un sous-arbre de* A_i'*. L'ensemble des n-uplets* $(A_1, \ldots, A_n)$ *muni de la relation* $\prec$ *représente un ordre partiel.*
Considérons $d(S_{\Gamma_i}^{N_i,A_i}(M_i))$*, la preuve du prédicat* $S_{\Gamma_i}^{N_i,A_i}(M_i)$*, qui est un arbre. Nous réaliserons notre induction sur l'ensemble des n-uplets*
$(d(S_{\Gamma_1}^{N_1,A_1}(M_1)), \ldots, (d(S_{\Gamma_n}^{N_n,A_n}(M_n)))$.
Pour les preuves que nous aurons à effectuer, le cas de base pour cette induction se situe lorsque l'ensemble des M_i *pour* $1 \leq i \leq n$*, sont des termes en forme normale.*

Démonstration. Par induction sur les dérivations, voyons le cas des règles suivantes :

- (S-App).
  Prouvons d'abord
  $S_\Gamma^C([M_2/x]B_2)$, $S_\Gamma^{N_1,(x:B_1)B_2}(M_1)$ et
  $S_\Gamma^{N_2,B_1}(M_2) \Longrightarrow S_\Gamma^{N_1(N_2),C}(M_1(M_2))$. $M_1(M_2)$ étant en forme normale de

tête faible.
Pour cela nous devons montrer que :

(a) $\Gamma \vdash^S M_1(M_2) \xrightarrow{nf} N_1(N_2) : C$,

(b) $M_1(M_2) \rhd M'$ implique que $S_\Gamma^{N_1(N_2),C}(M')$ et

(c) pour tout M_i sous-termes de $M_1(M_2)$, il existe B et Q tels que $\Gamma, \Delta \vdash^S M_i \xrightarrow{nf} B : Q$.

- Montrons (a).
Par hypothèse, on sait qu'on a $S_\Gamma^C([M_2/x]B_2)$, $S_\Gamma^{N_1,(x:B_1)B_2}(M_1)$, $S_\Gamma^{N_2,B_1}(M_2)$ et que $M_1(M_2)$ est en forme normale de tête faible. Par conséquent on a $\Gamma \vdash M_1 \xrightarrow{nf} N_1 : (x : B_1)B_2$, $\Gamma \vdash M_2 \xrightarrow{nf} N_2 : B_1$, $\Gamma \vdash [M_2/x]B_2 \xrightarrow{nf} C$ et $M_1(M_2)$ est en forme normale de tête faible. En appliquant la règle (S-App) on obtient $\Gamma \vdash^S M_1(M_2) \xrightarrow{nf} N_1(N_2) : C$.

- Montrons (b) par une induction sur le n-uplet
$(d(S_\Gamma^C([M_2/x]B_2)), \quad d(S_\Gamma^{N_1,(x:B_1)B_2}(M_1)), \quad d(S_\Gamma^{N_2,B_1}(M_2)))$.

Deux cas sont à examiner :

  * $M_1 \rhd M_1'$, alors on a par définition $S_\Gamma^{N_1,(x:B_1)B_2}(M_1')$. Par hypothèse d'induction on trouve $S_\Gamma^{N_1(N_2),C}(M_1'(M_2))$. Nous pouvons, en effet, appliquer l'hypothèse d'induction car l'arbre $d(S_\Gamma^{N_1,(x:B_1)B_2}(M_1'))$ est un sous-arbre de $d(S_\Gamma^{N_1,(x:B_1)B_2}(M_1))$, de ce fait nous avons la relation d'ordre suivante :

$$(d(S_\Gamma^C([M_2/x]B_2)), d(S_\Gamma^{N_1,(x:B_1)B_2}(M_1')), d(S_\Gamma^{N_2,B_1}(M_2))) \prec$$
$$(d(S_\Gamma^C([M_2/x]B_2)), d(S_\Gamma^{N_1,(x:B_1)B_2}(M_1)), d(S_\Gamma^{N_2,B_1}(M_2))).$$

  * $M_2 \rhd M_2'$, alors :

- $\cdot$ si $x \in FV(B_2)$, on a $[M_2/x]B_2 \triangleright [M_2'/x]B_2$, de même que $S_\Gamma^{N_2,B_1}(M_2')$, par hypothèse d'induction on peut conclure que $S_\Gamma^{N_1(N_2),C}(M_1(M_2'))$;

- $\cdot$ si x $\notin FV(B_2)$, on a juste $S_\Gamma^{N_2,B_1}(M_2')$, par hypothèse d'induction on peut conclure que $S_\Gamma^{N_1(N_2),C}(M_1(M_2'))$.

– Montrons (c) :
Il a été montré en (a) que $\Gamma \vdash^S M_1(M_2) \xrightarrow{nf} N_1(N_2) : C$. Le lemme 3.31 nous permet de conclure.

Revenons à notre démonstration initiale. Par hypothèse d'induction on a $S_\Gamma^C([M_2/x]B_2)$, $S_\Gamma^{N_1,(x:B_1)B_2}(M_1)$ et $S_\Gamma^{N_2,B_1}(M_2)$, de plus nous savons que $M_1(M_2)$ est en forme normale de tête faible par hypothèse. On peut conclure que $S_\Gamma^{N_1(N_2),C}(M_1(M_2))$.

- (W-β).
Prouvons d'abord $S_\Gamma^C([M_2/x]B_2)$, $S_\Gamma^{N,(x:B_1)B_2}(x : A_1[M_0])$, $S_\Gamma^{N_1,B_1}(M_2)$ et $S_\Gamma^{P,C}([M_2/x]M_0) \Longrightarrow S_\Gamma^{P,C}(([x : A_1]M_0)(M_2))$.

Nous devons montrer :

(a) $\Gamma \vdash^S ([x : A_1]M_0)(M_2) \xrightarrow{nf} P : C$,

(b) $([x : A_1]M_0)(M_2) \triangleright V$ implique que $S_\Gamma^{P,C}(V)$ et

(c) pour tout M_i sous-termes de $([x : A_1]M_0)(M_2)$, il existe B et Q tels que $\Gamma, \Delta \vdash^S M_i \xrightarrow{nf} B : Q$.

– Montrons (a) :
Par hypothèse on a $S_\Gamma^C([M_2/x]B_2)$, $S_\Gamma^{N,(x:B_1)B_2}(x : A_1[M_0])$ et $S_\Gamma^{N_1,B_1}(M_2)$, on en déduit $\Gamma \vdash^S [M_2/x]B_2 \xrightarrow{nf} C$, $\Gamma \vdash^S x : A_1[M_0] \xrightarrow{nf} N : (x : B_1)B_2$ et $\Gamma \vdash^S M_2 \xrightarrow{nf} N_1 : B_1$. En appliquant la règle (W-β) on trouve $\Gamma \vdash^S ([x : A_1]M_0)(M_2) \xrightarrow{wh} [M_2/x]M_0 : C$. De plus, on sait par hypo-

thèse que $S_\Gamma^{P,C}([M_2/x]M_0)$, alors $\Gamma \vdash^S [M_2/x]M_0 \xrightarrow{nf} P : C$. La règle (S-WH) nous permet de conclure.

– Montrons (b) par une induction sur le n-uplet :
$(d(S_\Gamma^C([M_2/x]B_2)), d(S_\Gamma^{N,(x:B_1)B_2}(x : A_1[M_0])), d(S_\Gamma^{N_1,B_1}(M_2), d(S_\Gamma^{P,C}([M_2/x]M_0)))$.

Soit V tel que $([x : A_1]M_0)(M_2) \triangleright V$. Examinons les cas suivants.

∗ $V \equiv [M_2/x]M_0$, par hypothèse on a $S_\Gamma^{P,C}([M_2/x]M_0)$.

∗ $V \equiv ([x : A_1']M_0)(M_2)$ avec $A_1 \triangleright A_1'$. Dans ce cas $[x : A_1]M_0 \triangleright [x : A_1']M_0$, par définition on a $S_\Gamma^{N,(x:B_1)B_2}(x : A_1'[M_0])$, on peut conclure par hypothèse d'induction que $S_\Gamma^{P,C}(([x : A_1']M_0)(M_2))$.

∗ $V \equiv ([x : A_1]M_0')(M_2)$ avec $M_0 \triangleright M_0'$. Dans ce cas $[x : A_1]M_0 \triangleright [x : A_1]M_0'$ et $[M_2/x]M_0 \triangleright [M_2/x]M_0'$ d'où $S_\Gamma^{N,(x:B_1)B_2}(x : A_1[M_0'])$ et $S_\Gamma^{P,C}([M_2/x]M_0')$, on peut conclure par hypothèse d'induction.

∗ $V \equiv ([x : A_1]M_0)(M_2')$ avec $M_2 \triangleright M_2'$. Dans ce cas $[M_2/x]B_2 \triangleright [M_2'/x]B_2$ et $[M_2/x]M_0 \triangleright [M_2'/x]M_0$, on a donc $S_\Gamma^C([M_2'/x]B_2)$, $S_\Gamma^{N_1,B_1}(M_2')$ et $S_\Gamma^{P,C}([M_2'/x]M_0)$ on peut conclure.

– Montrons (c).
Il a été montré en (a) que $\Gamma \vdash^S ([x : A_1]M_0)(M_2) \xrightarrow{nf} P : C$. Le lemme 3.31 nous permet de conclure.

Revenons à notre démonstration initiale. Par hypothèse d'induction on a : $S_\Gamma^C([M_2/x]B_2)$, $S_\Gamma^{N,(x:B_1)B_2}(x : A_1[M_0])$ et $S_\Gamma^{N_1,B_1}(M_2)$. Par hypothèse on a $S_\Gamma^{P,C}([M_2/x]M_0)$. On obtient $S_\Gamma^{P,C}(([x : A_1]M_0)(M_2))$.

- (W-ϑ).
Prouvons d'abord que $S_\Gamma^{P_1,(F_m)A}(E[\bar{\Theta}_m](C, a_1, ..., a_m))$,
$S_\Gamma^{P_2,(F_n)F_m}(E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m]))$,
$S_\Gamma^{P_4,A}(E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k))$ et $S_\Gamma^{v,F_n}(k) \Longrightarrow$
$S_\Gamma^{P_4,A}(E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m])(k))$.

Pour cela nous devons montrer que :

(a) $\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) \xrightarrow{nf} P_4 : A$,

(b) $E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) \rhd V$ implique que $S^{P_4,A}_\Gamma(V)$ et

(c) pour tout M_i sous-termes de $E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)$, il existe B et Q tels que $\Gamma, \Delta \vdash^S M_i \xrightarrow{nf} B : Q$.

– Montrons (a).
Par hypothèse on a :
$S^{P_1,(F_m)A}_\Gamma(E[\bar{\Theta}_m](C, a_1, ..., a_m)), S^{P_2,(F_n)F_m}_\Gamma(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]))$ et $S^{v,F_n}_\Gamma(k)$, on en déduit $\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m) \xrightarrow{nf} P_1 : (F_m)A$, $\Gamma \vdash^S E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) \xrightarrow{nf} P_2 : (F_n)F_m$, $\Gamma \vdash^S k \xrightarrow{nf} v : F_n$. En appliquant la règle (W-ϑ) on obtient :

$$\Gamma \vdash^S E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) \xrightarrow{wh} E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A.$$

De plus, on sait par hypothèse que $S^{P_4,A}_\Gamma(E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k))$, alors la règle (S-WH) nous permet de conclure que $\Gamma \vdash^S E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) \xrightarrow{nf} P_4 : A$.

– Montrons (b), par une induction sur le n-uplet :
$(d(S^{P_1,(F_m)A}_\Gamma(E[\bar{\Theta}_m](C, a_1, ..., a_m))), \quad d(S^{P_2,(F_n)F_m}_\Gamma(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]))), d(S^{v,F_n}_\Gamma(k)), \quad d(S^{P_4,A}_\Gamma(E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k))))$.

Soit V tel que
$E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) \rhd V$,
par définition de la règle (W-ϑ) nous savons que les termes $E[\bar{\Theta}_m](C, a_1, ..., a_m)$ et $E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])$ sont déjà en forme normale.
Les cas suivants sont à examiner.

* $V \equiv E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k)$, par hypothèse on a $S_\Gamma^{P_4,A}(E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k))$, on peut conclure.

* $V \equiv E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k'))$ avec $k \triangleright k'$. Dans ce cas $E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) \triangleright E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k')$. Par hypothèse et par définition on a $S_\Gamma^{v,F_n}(k')$ et $S_\Gamma^{P_4,A}(E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k'))$. On peut appliquer l'hypothèse d'induction et conclure par $S_\Gamma^{P_4,A}(E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k')))$.

– Montrons (c) :
  Il a été montré en (a) que $\Gamma \vdash^S E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) \xrightarrow{nf} P_4 : A$.
  Le lemme 3.31 nous permet de conclure.

Revenons à notre démonstration initiale. Par hypothèse d'induction on a : $S_\Gamma^{P_1,(F_m)A}(E[\bar{\Theta}_m](C, a_1, ..., a_m))$, $\quad S_\Gamma^{P_2,(F_n)F_m}(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]))$ et $S_\Gamma^{v,F_n}(k)$. Par hypothèse, on a $S_\Gamma^{P_4,A}(E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k))$. On obtient finalement $S_\Gamma^{P_4,A}(E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))$.

□

Tous les résultats importants concernant les réductions sont les corollaires des lemmes 3.32, 3.26, et 3.29, qui suivent.

Corollaire 3.34 (Préservation du type pour UTT^{rS}). *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$ *et* $M \triangleright^* N$*, alors* $\Gamma \vdash^S N \xrightarrow{nf} P : A$.

Corollaire 3.35 (Normalisation forte pour UTT^{rS}). *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$*, alors* M *est fortement normalisant.*

Corollaire 3.36 (Confluence pour UTT^{rS}). *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$*,* $M \triangleright^* M_1$ *et* $M \triangleright^* M_2$*, alors* $M_1 \triangleright^* P$ *et* $M_2 \triangleright^* P$.

Les propriétés de normalisation forte, de préservation du type et de confluence sont démontrées pour UTT^{rS}. Toujours en suivant le procédé de Goguen, il faut maintenant transférer ces propriétés au calcul UTT^{r} en prouvant que UTT^{r} est correcte par rapport à UTT^{rS}.

3.2.4 Construction d'un modèle ensembliste classique pour UTT^{r}

Dans cette section, toujours en suivant la preuve réalisée par Goguen, nous construisons un modèle ensembliste classique pour UTT^{r} afin de proposer un nouveau principe d'induction nécessaire pour prouver que UTT^{rS} est correcte par rapport à UTT^{rS}. Pour plus de détails concernant la construction de ce modèle ensembliste, nous renvoyons le lecteur à la section A.2 de l'annexe.

Par rapport à la démarche de Goguen, le seul lemme dont la preuve est modifiée par l'ajout de la règle ϑ-eq est le lemme suivant.

Lemme 3.37 (Correction de UTT^{r-} par rapport au modèle ensembliste)**.** *Pour tout jugement $\Gamma \vdash^{-} J$ dérivable, $[|\Gamma|]$ est définie, de plus :*

- *si le jugement $\Gamma \vdash^{-} A$**kind** est dérivable et $\rho \in [|\Gamma|]$, alors $[|A|](\rho)$ est définie ;*
- *si le jugement $\Gamma \vdash^{-} M : A$ est dérivable et $\rho \in [|\Gamma|]$, alors $[|A|](\rho)$ et $[|M|](\rho)$ sont définies et $[|M|](\rho) \in [|A|](\rho)$;*
- *si le jugement $\Gamma \vdash^{-} A = B$ est dérivable et $\rho \in [|\Gamma|]$, alors $[|A|](\rho)$ et $[|B|](\rho)$ sont définies et $[|A|](\rho) = [|B|](\rho)$;*
- *si le jugement $\Gamma \vdash^{-} M = N : A$ est dérivable et $\rho \in [|\Gamma|]$, alors $[|M|](\rho)$, $[|N|](\rho)$ et $[|A|](\rho)$ sont définies, $[|M|](\rho) = [|N|](\rho)$, $[|M|](\rho) \in [|A|](\rho)$ et $[|N|](\rho) \in [|A|](\rho)$.*

Rappel : pour un environnement d'ensemble ρ, l'interprétation d'un terme M et d'une sorte A est notée $[|M|](\rho)$ et $[|A|](\rho)$ respectivement (voir la section A.2 de *l'annexe).*

Démonstration. Par induction sur les dérivations de UTT^{r-}. La considération de la nouvelle règle ϑ-eq ne présente aucune difficulté particulière par rapport à la preuve initiale réalisée par Goguen.

□

3.2.5 UTT^r est correcte par rapport à UTT^{rS}

La preuve est faite dans cette section que UTT^r est correcte par rapport à UTT^{rS}. Nous introduisons au préalable les notions et propriétés nécessaires à la compréhension des lemmes dont la modification était inéluctable.

3.2.5.1 Objets sémantiques

Définition 3.38 (Ensemble valué)**.** *On définit ici les objets sémantiques pour un contexte Δ et une sorte A, $SO_\Delta(A)$, les ensembles valués $V(\Delta \vdash^S M : A)$ et les ensembles saturés $SAT_\Delta(A)$ simultanément où nous présumons que $\Delta \vdash^S A \xrightarrow{nf} A$ ou $\Delta \vdash^S M \xrightarrow{nf} M : A$. Les définitions suivantes sont valables pour une sorte A arbitraire en forme normale.*

- *Un objet sémantique est une paire (M, v), où $\Delta \vdash^S M \xrightarrow{nf} P : A$ et v est une famille d'objets $v(\delta) \in V(\Delta' \vdash^S \widehat{\delta}(M) : \widehat{\delta}(M))$, indexés par des renommages δ de Δ' vers Δ.*

- *$SO_\Delta(A)$ est l'ensemble des objets sémantiques pour Δ et A.*

- *S est un ensemble saturé pour Δ et A,* notation *$SAT_\Delta(A)$, si S est un ensemble d'objets sémantiques pour Δ et A tels que :*

  - *(S_1) Si M est un terme de base tel que $\Delta \vdash^S M \xrightarrow{nf} P : A$ alors $(M, \lambda\delta.*)$ est dans S, et*

  - *(S_2) Si (N, v) est dans S et $\Delta \vdash^S M \xrightarrow{wh} N : A$ alors (M, v) est dans S.*

- *Une famille $v(\delta)$ d'ensembles indexés par des renommages δ de Δ' vers Δ est famille saturée d'ensembles pour Δ et A si $v(\delta)$ est un ensemble saturé pour Δ' et A pour chaque renommage δ de Δ' vers Δ.*

La définition de $V(\Delta \vdash^S M : A)$, où $\Delta \vdash^S M \xrightarrow{nf} M : A$ est donnée par induction sur la complexité de A (voir la section A.4 de l'annexe).

- $V(\Delta \vdash^S M : A) =_{df} \{*\}$ *si M est un terme de base, pour tout A.*
- $V(\Delta \vdash^S \Lambda(A, P, g) : El(Prf(\forall(A, P))))) =_{df} \{*\}$.
- $V(\Delta \vdash^S \forall(A, P) : El(Prop)) =_{df} SAT_\Delta(El(Prf(\forall(A, P)))))$.
- $V(\Delta \vdash^S \iota_i^X[\bar{\Theta}](N_1, \ldots, N_n) : El(M^X[\bar{\Theta}])) =_{df} f_\Delta(\Theta_i, N_1, \ldots, N_n)$*, où $f_\Delta(\Theta, N_1, \ldots, N_n)$ est définie par induction sur la complexité de $M^X[\bar{\Theta}]$ et $SCH^S_{\Gamma;X}(\Theta_i)$:*
  - $f_\Delta(X, ()) =_{df} \{*\}$,
  - $f_\Delta((x : A)\Theta_0, N_1, \ldots, N_n) =_{df} V(\Delta \vdash^S N_1 : A) \times f_\Delta(\Theta_0', N_2, \ldots, N_n)$, *où* $\Delta \vdash^S [N_1/x]\Theta_0 \xrightarrow{nf} \Theta_0'$.
  - $f_\Delta((\Phi)\Theta_0, N_1, \ldots, N_n) =_{df} V(\Delta \vdash^S N_1 : \Phi(M^X[\bar{\Theta}])) \times f_\Delta(\Theta_0, N_2, \ldots, N_n)$.
- $V(\Delta \vdash^S prop : El(U)) =_{df} SAT_\Delta(El(Prop)))$.
- $V(\Delta \vdash^S prf(\forall(A, P)) : El(U)) =_{df} SAT_\Delta(El(Prf(\forall(A, P)))))$.
- $V(\Delta \vdash^S \mu^X[\bar{\Theta}] : El(U)) =_{df} SAT_\Delta(El(M^X[\bar{\Theta}])))$.
- $V(\Delta \vdash^S M :$ ***Type*** $=_{df} SAT_\Delta(El(M)))$ *si M n'est pas un terme de base.*
- $V(\Delta \vdash^S M : (x : A_1)A_2)$*, si M n'est pas un terme de base, est l'ensemble des f tel que :*
  - $dom(f) \in SAT_\Delta(A_1)$ *et*

– *$f(N, v_N) \in V(\Delta \vdash^S P : B_2)$, où $(N, v_N) \in dom(f)$, $\Delta \vdash^S M(N) \xrightarrow{nf} P : B_2$ et $\Delta \vdash^S [N/x]A_2 \xrightarrow{nf} B_2$.*

Lemme 3.39. *Pour toute sorte A telle que $\Delta \vdash^S A \xrightarrow{nf} B$, on a $SO_\Delta(B) \in SAT_\Delta(B)$.*

Démonstration. Par induction sur les dérivations.

□

Définition 3.40 (Monotonie)**.** *Si (M, v_M) est un objet sémantique pour Δ et A, et δ un renommage de Δ' vers Δ, alors :*

$$mon_\delta(M, v_M) =_{df} (\widehat{\delta}(M), \lambda\delta'.v_M(\delta' \circ \delta))$$

Par construction, on sait que $mon_\delta(M, v_M)$ est un objet sémantique pour Δ' et A.

Définition 3.41 (Application)**.** *Soit (M_1, v_1) un objet sémantique pour Δ et $(x : A_1)A_2$ tels que $\Delta \vdash^S M_1 \xrightarrow{nf} P_1 : (x : A_1)A_2$, et (M_2, v_2) est un objet sémantique pour Δ et A_1 tels que $\Delta \vdash^S M_2 \xrightarrow{nf} P_2 : A_1$. Nous montrons qu'il existe une sorte B_2 telle que $\Delta \vdash^S [M_2/x]A_2 \xrightarrow{nf} B_2$ et définissons :*

$$APP_\Delta((M_1, v_1), (M_2, v_2)) \in SO_\Delta(B_2)$$

la fonction partielle pour les applications sémantiques :

- *Si P_1 est un terme de base et $\Delta \vdash^S [M_2/x]A_2 \xrightarrow{nf} B_2$ alors*

  $$APP_\Delta((M_1, v_1), (M_2, v_2)) =_{df} (M_1(M_2), *)$$

  où on a $\Delta \vdash^S M_1(M_2) \xrightarrow{nf} P_1(P_2) : B_2$ par la règle (S-App) ;

- *Si P_1 n'est pas un terme de base, et $mon_\delta(M_2, v_2) \in dom(v_1(\delta))$ pour tout renommage δ de Δ' vers Δ, alors*

  $$APP_\Delta((M_1, v_1), (M_2, v_2)) =_{df} (M_1(M_2), \lambda\delta.v_1(\delta)(mon_\delta(M_2, v_2)))$$

où $\Delta \vdash^S [M_2/x]A_2 \xrightarrow{nf} B_2$ *et* $\Delta \vdash^S M_1(M_2) \xrightarrow{nf} P : B_2$ *par définition de* $v_2(id_\Delta) \in V(\Delta \vdash^S M_1 : (x : A_1)A_2)$.

Lemme 3.42. *On a l'égalité suivante :*
$APP''_\Delta(mon_{\delta'\circ\delta}(M, v_M), mon_{\delta'}(N, v_N)) = mon_{\delta'}(APP_{\Delta'}(mon_\delta(M, v_M), (N, v_N)))$, *avec* δ *un renommage de* Δ' *vers* Δ *et* δ' *un renommage de* Δ'' *vers* Δ'.

Démonstration. Avec les définitions de l'application et de la monotonie, et utilisant le lemme 3.23.

□

Notation
Nous notons $(M, v_M) \downarrow (N, v_N)$ où (M, v_M) et (N, v_N) sont des objets sémantiques pour Δ et A, s'il existe un terme P tel que $\Delta \vdash^S M \downarrow N[P] : A$ et $v_M = v_N$.

Définition 3.43 (Objets sémantiques uniformes)**.** *Soit* S *une famille saturée d'ensembles pour* Δ *et* A*.* S *est uniforme si* $(M, v_M) \in S(\delta)$ *implique que* (M, v_M) *est uniforme et* $mon_{\delta'}(M, v_M) \in S(\delta' \circ \delta)$ *pour tout renommage* δ' *de* Δ'' *vers* Δ'.

Un objet sémantique (M, v_M) *pour* Δ *et* A*, est uniforme si :*

- M *est un terme de base,*
- $M \equiv \forall(A, P)$ *et* v_M *est uniforme,*
- $M \equiv \Lambda(A, P, g)$*,*
- $M \equiv \iota_i^X[\bar{\Theta}](N_1, \ldots, N_n)$ *et* (N_i, v_i) *est uniforme pour* $1 \leq i \leq n$*,*
- $M \equiv prop$ *et* v_M *est uniforme,*
- $M \equiv prf(\forall(A, P))$ *et* v_M *est uniforme,*
- $M \equiv \mu^X[\bar{\Theta}]$ *et* v_M *est uniforme,*

- $A \equiv$ ***Type***, *M n'est pas un terme de base et v_M est uniforme, ou*

- $\Delta \vdash^S M \xrightarrow{nf} P : (x : A_1)A_2$ *et*

  1) $\lambda\delta.dom(v_M(\delta))$ *est uniforme,*

  2) si $(N, v_N) \in dom(v_M(\delta))$*, alors* $APP_{\Delta'}(mon_\delta(M, v_M), (N, v_N))$ *est uniforme, et*

  3) si $(N_1, v_1) \downarrow (N_2, v_2)$*, alors* $APP_{\Delta'}(mon_\delta(M, v_M), (N_1, v_1)) \downarrow APP_{\Delta'}(mon_\delta(M, v_M), (N_2, v_2))$.

Définition 3.44 (Objets sémantiques uniformes)**.** *L'ensemble des objets sémantiques uniformes pour Δ et A,* notation $USO_\Delta(A)$*, est l'ensemble des objets sémantiques pour Δ et A qui sont uniformes*

Lemme 3.45. *Nous listons quelques propriétés simples des objets sémantiques uniformes :*

- *$USO_\Delta(A)$ est uniforme et est dans $SAT_\Delta(A)$.*

- *Si (M, v_M) est un objet sémantique uniforme alors $mon_\delta(M, v_M)$ est aussi un objet sémantique uniforme, pour tout renommage δ de Δ' vers Δ.*

3.2.5.2 Interprétation

Définition 3.46 (Substitution)**.** *Une substitution ϕ de Δ vers Γ, où $\Gamma \equiv x_1 : A_1, \ldots, x_m : A_m$ et $\Delta \vdash^S$* ***valid****, est une pré-substitution sur Γ telle que :*

$$\Delta \vdash^S \widehat{\phi}(A_i) \xrightarrow{nf} B_i \text{ et } \Delta \vdash^S \phi(x_i) \xrightarrow{nf} P : B_i \text{ pour } 1 \leq i \leq m.$$

Une pré-substitution est un environnement pour termes.

Définition 3.47 (Valuation)**.** *Une valuation de* Δ *vers* Γ *est une pre-valuation telle que le terme constituant est une substitution de* Δ *vers* Γ *.*
Une pré-valuation est un environnement pour objets sémantiques dans un contexte donné.

Lemme 3.48. *Si* ρ *est une valuation de* Δ *vers* Γ *et* δ *un renommage de* Δ' *vers* Δ*, alors* $mon_\delta(\rho)$ *est une valuation de* Δ' *vers* Γ*.*

La valeur de la fonction d'interprétation pour un terme, relative à une valuation de Δ vers Γ, est un objet sémantique pour Δ. Cet objet sémantique est une paire composée d'un terme bien typé dans UTT^{rS} dans le contexte Δ et d'une famille de valeurs indexées par un renommage de Δ' vers Δ.
Nous renvoyons le lecteur à [Gog94] pour une définition plus précise d'une valuation.

Définition 3.49 (Interprétation)**.** *On définit l'interprétation d'un terme* M*, notée* $[|M|]\rho\Delta$*, et d'une sorte* A*, notée* $[|A|]\rho\Delta$*, par induction sur la structure de* M *et de* A*.*
$[|M|]\rho\Delta$ *est une paire* $(\hat{\phi}(M), V[|M|]\rho\Delta)$*, où* $V[|M|]\rho\Delta$ *est une famille de valeurs indexées par un renommage* δ *de* Δ' *vers* Δ*.*

- $V[|M|]\rho\Delta =_{df} val(x)$ *si* $x \in dom(\rho)$*.*
- $[|(x : A_1)A_2|]\rho\Delta$ *est définie si, pour* $x \notin dom(\rho)$ *:*
  - $\Delta \vdash^S \hat{\phi}(A_1) \xrightarrow{nf} B_1$*,*
  - $\Delta, y : \hat{\phi}(A_1) \vdash^S \phi\widehat{[x := y]}(A_2) \xrightarrow{nf} B_2$*,*
  - $[|A_1|]\rho\Delta$ *est définie, est uniforme, et est une famille saturée d'ensembles pour* Δ *et* B_1*, et*
  - *pour tout renommage* δ *de* Δ' *vers* Δ*, et pour tout* $(M_2, v_2) \in [|A_1|]\rho\Delta(\delta)$*, on a :*

* $\Delta' \vdash^S [M_2/x]B_2 \xrightarrow{nf} B_2'$,

* $[|A_2|]mon_\delta(\rho)[x := (M_2, v_2)]\Delta'$ *est définie, et est une famille saturée d'ensembles pour* Δ' *et* B_2', *où* $\Delta' \vdash^S \phi[\widehat{x := M}_2](A_2) \xrightarrow{nf} B_2'$.

$[|(x : A_1)A_2|]\rho\Delta$ *est l'ensemble des* (M_1, v_1) *tels que*

– (M_1, v_1) *est un objet sémantique uniforme pour* Δ' *et* $(x : B_1)B_2$ *et*

– *pour tout renommage* δ' *de* Δ' *vers* Δ *:*

  * *si* M_1 *est un terme de base, alors* $dom(v_1(\delta')) = [|A_1|]mon_{\delta'\circ\delta}(\rho)\Delta''(id_{\Delta''})$ *et*

  * *pour tout* $(M_2, v_2) \in [|A_1|]mon_{\delta'\circ\delta}(\rho)\Delta''(id_{\Delta''})$, *on a* $APP_{\Delta''}(mon_{\delta'}(M_1, v_1), (M_2, v_2)) \in [|A_2|]mon_{\delta'\circ\delta}(\rho)[x := (M_2, v_2)]\Delta''(id_{\Delta''})$.

- $[|M_1(M_2)|]\rho\Delta$ *est définie si :*

  – $[|M_1|]\rho\Delta$ *est définie et appartient à* $USO_\Delta((x : B_1)B_2)$, *où on a* $\Delta \vdash^S \hat{\phi}(M_1) \xrightarrow{nf} P_1 : (x : B_1)B_2$ *;*

  – $[|M_2|]\rho\Delta$ *est définie et appartient à* $USO_\Delta(B_1)$ *;*

  – *si* P_1 *est un terme de base alors on a* $\Delta \vdash^S [\hat{\phi}(M_2)/x]A_2 \xrightarrow{nf} P_1 : B_2'$ *et*

  – *si* P_1 *n'est pas un terme de base alors* $[|M_2|]\rho\Delta \in dom(V[|M_1|]\rho\Delta(id_\Delta))$.

  $[|M_1(M_2)|]\rho\Delta =_{df} APP_\Delta([|M_1|]\rho\Delta, [|M_2|]\rho\Delta)$.

- $[|M^X[\bar{\Theta}]|]\rho\Delta$ *est définie si on a* Δ, Y : ***Type*** $\vdash^S \phi[\widehat{X := Y}](\Theta_i) \xrightarrow{nf} \Theta_i'$ *pour* $X \notin dom(\rho)$, *et* $[|\Theta_i|]|]\rho'\Delta'$ *est définie pour toute valuation* ρ' *de* Δ' *vers* Δ, X : ***Type***.

$[|M^X[\bar{\Theta}]|]\rho\Delta(\delta)$ *est définie par induction transfinie sur le cardinal* α_0*, comme étant le plus petit ensemble saturé* S *tel que si :*

– $(M_j, v_j) \in [|M_j|]mon_\delta(\rho)[X := S][x_1 := (M_1, v_1)] \dots \Delta'(id_{\Delta'})$ *et*

– $\Delta', Y : \textbf{\textit{Type}} \vdash^S \phi[\widehat{X := Y}](\Theta_j) \downarrow \Theta'_j[\Theta''_j]$ *pour* $1 \leq j \leq n$ *alors*

$$(\iota_i^X[\bar{\Theta}'](N_1, \dots, N_n), \lambda\delta'. < v_1(\delta'), \dots, v_n(\delta') >) \in S.$$

- $[|\iota_i^X[\bar{\Theta}]|]\rho\Delta$ *est définie si on a* $\Delta, Y : \textbf{\textit{Type}} \vdash^S \phi[\widehat{X := Y}](\Theta_i) \xrightarrow{nf} \Theta'_i$ *pour* $X \notin dom(\rho)$*, et* $[|\Theta_i|]|\rho'\Delta'$ *est définie pour toute valuation* ρ' *de* Δ' *vers* $\Delta, X : \textbf{\textit{Type}}$*.*

  $V[|\iota_i^X[\bar{\Theta}]|]\rho\Delta(\delta) =_{df}$

  $\lambda(M_1, v_1) \in [|M_j|]mon_\delta(\rho)[X := [|M^X[\bar{\Theta}]|]\rho\Delta]\Delta'(id_{\Delta'})$
  $\dots$
  $\lambda(M_n, v_n) \in [|M_n|]mon_\delta(\rho)[X := [|M^X[\bar{\Theta}]|]\rho\Delta][x_1 := (M_1, v_1)] \dots \Delta'(id_{\Delta'}).$
  $< v_1(id_{\Delta'}), \dots, v_n(id_{\Delta'}) >.$

- $[|E^X[\bar{\Theta}]|]\rho\Delta$ *est définie si on a* $\Delta, Y : \textbf{\textit{Type}} \vdash^S \phi[\widehat{X := Y}](\Theta_i) \xrightarrow{nf} \Theta'_i$ *pour* $X \notin dom(\rho)$*, et* $[|\Theta_i|]|\rho'\Delta'$ *est définie pour toute valuation* ρ' *de* Δ' *vers* $\Delta, X : \textbf{\textit{Type}}$*.*

  Nous présumons que

  $(C, v_C) \in [|(X)\textbf{\textit{Type}}|]mon_\delta(\rho)[X := [|M^X[\bar{\Theta}]|]mon_\delta(\rho)\Delta']\Delta'(id_{\Delta'}),$
  $(f_1, v_{f_1}) \in [|\Theta_1^\circ[A, C, z]|]\rho'_1\Delta'(id_{\Delta'})$
  $\dots$
  $(f_n, v_{f_n}) \in [|\Theta_n^\circ[A, C, z]|]\rho'_n\Delta'(id_{\Delta'})$

  où
  $\rho'_i =_{df} mon_\delta(\rho)[A := [|M^X[\bar{\Theta}]|]mon_\delta(\rho)\Delta'][C := (C, v_C)][z := [|\iota_i^X[\bar{\Theta}]|]mon_\delta(\rho)\Delta']$
  pour $1 \leq i \leq n$*.*

On a $V[|E^X[\bar{\Theta}]|]\rho\Delta(\delta)((C, v_C), (f_1, v_1), \ldots, (f_n, v_{f_n})) =_{df} R^\alpha$*, la fonction partielle définie par induction transfinie sur la complexité de* $M^X[\bar{\Theta}]$ *:*

– $\alpha = 0$*, alors*

$$R^0 =_{df} \lambda(M, v_M) \in [|El(M^X[\bar{\Theta}])|]mon_\delta(\rho)\Delta'(id_{\Delta'}).$$

$$\begin{cases} * & \textit{si } M \textit{ est un terme de base} \\ R^0(N, V_M) & \textit{si } \Delta' \vdash^S M \xrightarrow{wh} N : El(M^X[\bar{\Theta}]) \end{cases}$$

– $\alpha = succ(\beta)$*. Soit* $(M, v_M) \in [|El(M^X[\bar{\Theta}])|]mon_\delta(\rho)\Delta'(id_{\Delta'})$*, on a*

  * $R^{succ(\beta)}(M, v_M) =_{df} *$ *si* M *est un terme de base ;*

  * $R^{succ(\beta)}(M, v_M) =_{df} R^{succ(\beta)}(N, v_M)$ *si* $\Delta' \vdash^S M \xrightarrow{wh} N : El(M^X[\bar{\Theta}])$ *où* $R^{succ(\beta)}(N, v_M)$ *est définie ; et*

  * *soit* $\Phi_{i_1}, \ldots, \Phi_{i_j}$*, les opérateurs strictement positifs de* Θ*, et définissons* $\Phi^*_{i_k}$ *l'interprétation de* $\Phi^\natural_{i_k}[A, C, f, z]$ *sous la valuation* $\rho[A := [|M^X[\bar{\Theta}]|]\rho\Delta][C := (C, v_C)][f := R^\beta][z := (N_{i_k}, v_{i_k})]$*, pour* $1 \leq k \leq j$*. Alors on a*
    $R^{succ(\beta)}(M, v_M) =_{df} v_{f_i}(N_1, \ldots, N_n, \Phi^*_{i_1}, \ldots, \Phi^*_{i_j})$ *si* $M \equiv \iota^X_i[\bar{\Theta}'](N_1, \ldots, N_n)$ *et* $\Phi^*_{i_k} \in [|\Phi^\circ[A, C, z]|]\rho[A := [|M^X[\bar{\Theta}]|]\rho\Delta][C := (C, v_C)][z := (N_{i_k}, v_{i_k})]\Delta$ *pour* $1 \leq k \leq j$*.*

– *Limit* α*, alors* $R^\alpha(M, v_M) =_{df} R^\beta(M, v_M)$*, où* $\beta =_{df} \bigcap\{\gamma \in \alpha \,|\, M \in [|M^X[\bar{\Theta}]|]\rho\Delta^\gamma(\delta)\}$.

Remarque 3.50 (Interprétation des types finis)**.** *Nous définissons ici l'interprétation dans le cas des types finis. Cette interprétation n'est autre que le cas particulier de l'interprétation des types inductifs définie par Goguen dans [Gog94].*

Considérons le type fini $F_n =_{df} M[<\underbrace{X, \ldots, X}_{n fois}>]$*, où on a les constructeurs :*

$1_{F_n} : F_n, \ldots, n_{F_n} : F_n$.

- $[|F_n|]_\rho\Delta$ *est définie comme étant l'ensemble saturé contenant les éléments canoniques suivants :* $(1_{F_n}, \lambda\delta.*), \ldots, (n_{F_n}, \lambda\delta.*)$ *;*

- *de plus on a :* $[|i_{F_n}|]_\rho\Delta =_{df} (i_{F_n}, \lambda\delta.*)$.

Notons que par souci de concision et afin de mettre l'accent sur les modifications que nous avons portées à la preuve initiale, nous avons défini l'interprétation uniquement pour les termes et les sortes qui serviront à la compréhension de notre preuve.

Lemme 3.51 (Interprétation bien définie)**.** *Soit ρ une valuation de Δ vers Γ, alors :*

- *si $[|M|]\rho\Delta$ est définie alors il existe une sorte B telle que $[|M|]\rho\Delta \in SO_\Delta(B)$;*

- *Si $[|A|]\rho\Delta$ est définie alors $\Gamma \vdash^S A \xrightarrow{nf} B$ est dérivable et $[|A|]\rho\Delta \in SAT_\Delta(B)$.*

Si δ est un renommage de Δ' vers Δ alors :

- *si $[|A|]\rho\Delta$ est définie alors $mon_\delta([|A|]\rho\Delta) = [|A|]mon_\delta(\rho)\Delta'$;*

- *si $[|M|]\rho\Delta$ est définie alors $mon_\delta([|M|]\rho\Delta) = [|M|]mon_\delta(\rho)\Delta'$.*

Démonstration. Par induction simultanée sur la structure de M et de A. □

Définition 3.52 (Interprétation de contexte)**.** *L'interprétation d'un contexte Γ, notée $[|\Gamma|]\Delta$, est l'ensemble des valuations de Δ vers Γ, définies par induction sur la structure de Γ :*

- $[| \quad |]\Delta =_{df} \epsilon$ *;*

- $[|\Gamma, x : A|]\Delta =_{df} \{\rho[x := (M, v_M)] \mid \rho \in [|\Gamma|]\Delta \wedge (M, v_M) \in [|A|]\rho\Delta(id_\Delta)\}$, *si* $[|\Gamma|]\Delta$ *est définie, et* $[|A|]\rho\Delta$ *est définie pour tout* $\rho \in [|\Gamma|]\Delta$.

3.2.5.3 Correction

Le lemme suivant représente le résultat le plus important de cette section qui va nous permettre de prouver que UTT^r est correcte par rapport à UTT^{rS}.

Lemme 3.53 (Correction)**.** *Pour tout jugement* $\Gamma \vdash J$ *dérivable,* $[|\Gamma|]\Delta$ *est définie. De plus :*

- *si* $\Gamma \vdash A$***Kind*** *est dérivable et* $\rho \in [|\Gamma|]\Delta$*, alors* $[|A|]_\rho\Delta$ *est définie ;*

- *si* $\Gamma \vdash M : A$ *est dérivable et* $\rho \in [|\Gamma|]\Delta$*, alors* $[|M|]_\rho\Delta$ *et* $[|A|]_\rho\Delta$ *sont définies et* $[|M|]_\rho\Delta \in [|A|]_\rho\Delta$ *;*

- *si* $\Gamma \vdash A = B$ *est dérivable et* $\rho \in [|\Gamma|]\Delta$*, alors* $[|A|]_\rho\Delta$ *et* $[|B|]_\rho\Delta$ *sont définies et* $[|A|]_\rho\Delta \downarrow [|B|]_\rho\Delta$ *;*

- *si* $\Gamma \vdash M = N : A$ *est dérivable et* $\rho \in [|\Gamma|]\Delta$*, alors* $[|M|]_\rho\Delta$*,* $[|N|]_\rho\Delta$ *et* $[|A|]_\rho\Delta$ *sont définies,* $[|M|]_\rho\Delta \in [|A|]_\rho\Delta$ *et* $[|M|]_\rho\Delta \downarrow [|N|]_\rho\Delta$.

Démonstration. Preuve par induction sur les dérivations.
Nous considérons le cas de la règle (ϑ-eq) (règle rajoutée).
Prenons $\rho \in [|\Gamma|]\Delta$, et δ' un renommage de Δ' vers Δ.
Par hypothèse d'induction $[|(F_m)A|]_\rho\Delta$, $[|(F_n)F_m|]_\rho\Delta$ sont définies, il existe donc des sortes A', F'_n, et F'_m telles que $\Delta \vdash^S \widehat{\phi}(F_n) \xrightarrow{nf} F'_n$, $\Delta \vdash^S \widehat{\phi}(F_m) \xrightarrow{nf} F'_m$ et $\Delta \vdash^S \widehat{\phi}(A) \xrightarrow{nf} A'$.

Il nous faut d'abord montrer que
$[|A|]_\rho\Delta$, $[|E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta$ et $[|E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k)|]_\rho\Delta$ sont définies.

Commençons par $[|A|]_\rho\Delta$.
Par hypothèse d'induction, on sait que $[|(F_m)A|]_\rho\Delta$ est définie ce qui implique

que $[|A|]_\rho\Delta$ est définie par définition.

Afin de montrer que $[|E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)|]_\rho\Delta$ est définie, il faut pour cela prouver les points suivants.

- $[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta$ est définie et appartient à $USO_\Delta((F'_m)A')$, avec $\Delta \vdash^S \hat{\phi}(E[\bar{\Theta}_m](C, a_1, ..., a_m)) \xrightarrow{nf} P_1 : (F'_m)A'$.
  Par hypothèse d'induction $[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta$ et $[|(F_m)A|]_\rho\Delta$ sont définies et $[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta \in [|(F_m)A|]_\rho\Delta$.
  On a $\Delta \vdash^S \hat{\phi}(E[\bar{\Theta}_m](C, a_1, ..., a_m)) \xrightarrow{nf} P_1 : (F'_m)A'$, et par definition de $[|(F_m)A|]_\rho\Delta$, on sait que $[|(F_m)A|]_\rho\Delta \subseteq USO_\Delta((F'_m)A')$. Par conséquent, $[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta \in USO_\Delta((F'_m)A')$.

- $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)|]_\rho\Delta$ est définie et appartient à $USO_\Delta(F'_m)$.
  Pour prouver que $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)|]_\rho\Delta$ est définie, il nous faut montrer les points suivants.

  – $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta$ est définie et appartient à $USO_\Delta((F'_n)F'_m)$ avec $\Delta \vdash^S \hat{\phi}(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])) \xrightarrow{nf} P'_1 : (F'_n)F'_m$.
    Par hypothèse d'induction $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta$ et $[|(F_n)F_m|]_\rho\Delta$ sont définies et $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta \in [|(F_n)F_m|]_\rho\Delta$.
    On a $\Delta \vdash^S \hat{\phi}(E[\bar{\Theta}_m](C, a_1, ..., a_m)) \xrightarrow{nf} P'_1 : (F'_n)F'_m$, et par definition de $[|(F_n)F_m|]_\rho\Delta$, $[|(F_n)F_m|]_\rho\Delta \subseteq USO_\Delta((F'_n)F'_m)$. Par conséquent $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta \in USO_\Delta((F'_n)F'_m)$.

  – $[|k|]_\rho\Delta$ est définie et appartient à $USO_\Delta(F'_n)$.
    Par hypothèse d'induction, on sait que $[|k|]_\rho\Delta$ et $[|F_n|]_\rho\Delta$ sont définies et $[|k|]_\rho\Delta \in [|F_n|]_\rho\Delta$. Par définition de $[|F_n|]_\rho\Delta$, F_n étant un type fini, et par définition des objets sémantiques uniformes, on a bien $[|F_n|]_\rho\Delta \subseteq USO_\Delta(F'_n)$. Par conséquent $[|k|]_\rho\Delta \in USO_\Delta(F'_n)$.

  – $[|k|]_\rho\Delta \in dom(V[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta(id_\Delta))$, car P_1 n'est pas un terme de base.
    Par définition $dom(V[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta(id_\Delta)) = [|F_n|]_\rho\Delta$, or $[|k|]_\rho\Delta \in [|F_n|]_\rho\Delta$. On a donc $[|k|]_\rho\Delta \in dom(V[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta(id_\Delta))$.

On obtient donc que
$[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)|]_\rho\Delta =$
$APP_\Delta([|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta, [|k|]_\rho\Delta)$. Sachant que
$[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta \in [|(F_n)F_m|]_\rho\Delta, [|k|]_\rho\Delta \in [|F_n|]_\rho\Delta$
et par définition de $[|(F_n)F_m|]_\rho\Delta$, nous savons que
$APP_\Delta([|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta, [|k|]_\rho\Delta) \in [|F_m|]_\rho\Delta$. Comme
$[|F_m|]_\rho\Delta \subseteq USO_\Delta(F'_m)$ alors $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta \in USO_\Delta(F'_m)$.

- $[|(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta \in dom(V[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta(id_\Delta))$.
  Par définition $dom(V[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta(id_\Delta)) = [|F_m|]_\rho\Delta$, or nous savons que
  $[|(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta =_{df}$
  $APP_\Delta([|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])|]_\rho\Delta, [|k|]_\rho\Delta) \in [|F_m|]_\rho\Delta$, d'où
  $[|(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta \in dom(V[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta(id_\Delta))$.

On montre à l'aide entre autre, de la définition de l'interprétation pour les récurseurs que $[|E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k)|]_\rho\Delta$ est également définie.

Montrons maintenant que $[|E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta \downarrow [|E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k)|]_\rho\Delta$.
Par définition et en appliquant le lemme 3.51, on a
$[|E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta =_{df}$
$APP_\Delta([|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta\ ,\ [|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta) =_{df}$

$(\widehat{\phi}(E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))\ ,$
$\lambda\delta.V[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta(\delta)(mon_\delta([|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta))) =_{df}$

$(\widehat{\phi}(E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))\ ,$
$\lambda\delta.V[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta(\delta)([|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]mon_\delta(\rho)\Delta'))$.

On a bien $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]mon_\delta(\rho)\Delta' \in [|F_m|]mon_\delta(\rho)\Delta'$,
car $[|E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)|]_\rho\Delta \in [|F_m|]_\rho\Delta$.
De plus F_m étant un type fini, sa complexité est égale à 1 et $V[|E[\bar{\Theta}_m](C, a_1, ..., a_m)|]_\rho\Delta(\delta) =_{df} R^1$.

Ce qui donne :
$V[|E[\bar{\Theta}_m](C,a_1,...,a_m)|]_\rho\Delta(\delta)([|E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]mon_\delta(\rho)\Delta') =$

$$\begin{cases} * & \text{si } \Delta' \vdash^S \widehat{\delta\circ\phi}(k) \xrightarrow{nf} q : K\text{, où } q \text{ est un terme de base.} \\ v_{a_i} & \text{si } \Delta' \vdash^S \widehat{\delta\circ\phi}(k) \xrightarrow{nf} q : K\text{, où } q \text{ est un constructeur.} \end{cases}$$

On a donc :
(1) $[|E[\bar{\Theta}_m](C,a_1,...,a_m)(E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta =$
$(\widehat{\phi}(E[\bar{\Theta}_m](C,a_1,...,a_m)(E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))\ ,\ \lambda\delta.*)$, dans le cas où q est un terme de base.
Dans le cas où q est un constructeur, on a :
(1$'$) $[|E[\bar{\Theta}_m](C,a_1,...,a_m)(E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))|]_\rho\Delta =$
$(\widehat{\phi}(E[\bar{\Theta}_m](C,a_1,...,a_m)(E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))\ ,$
$\lambda\delta.V[|E[\bar{\Theta}_m](C,a_1,...,a_m)|]_\rho\Delta(\delta)(\widehat{\delta\circ\phi}(E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)\ ,\ \lambda\delta.v_{b_{\sigma(i)}})) =$
$(\widehat{\phi}(E[\bar{\Theta}_m](C,a_1,...,a_m)(E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k))\ ,\ \lambda\delta.v_{a_{\sigma(i)}})$.
En effet, si $\Delta' \vdash^S \widehat{\delta\circ\phi}(k) \xrightarrow{nf} i_{F_n} : F_n$ alors
$\Delta' \vdash^S \widehat{\delta\circ\phi}(E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) \xrightarrow{nf} b_{\sigma(i)} : F_m$.

De même on a :
(2) $[|E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})(k)|]_\rho\Delta =_{df} APP_\Delta([|E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})|]_\rho\Delta\ ,\ [|k|]_\rho\Delta) =$
$(\widehat{\phi}(E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})(k))\ ,\ \lambda\delta.V[|E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})|]_\rho\Delta(\delta)(mon_\delta([|k|]_\rho\Delta)))$=
$(\widehat{\phi}(E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})(k))\ ,\ \lambda\delta.V[|E[\bar{\Theta}_m](C,a_1,...,a_m)|]_\rho\Delta(\delta)([|k|]mon_\delta(\rho)\Delta')) =$
d'où $[|E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})(k)|]_\rho\Delta =$
$(\widehat{\phi}(E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})(k))\ ,\ \lambda\delta.*)$ si $\Delta' \vdash^S \widehat{\delta\circ\phi}(k) \xrightarrow{nf} q : K$, où q est un terme de base.

Dans le cas où q est un constructeur, on a :
(2$'$) $(\widehat{\phi}(E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})(k))\ ,\ \lambda\delta.v_{a_{\sigma(i)}})$ si $\Delta' \vdash^S \widehat{\delta\circ\phi}(k) \xrightarrow{nf} i_{F_n} : F_n$.

Nous pouvons conclure, à partir des égalités (1), (2) et (1$'$), (2$'$) que :
$[|E[\bar{\Theta}_m](C,a_1,...,a_m)(E[\bar{\Theta}_n](C_1,\iota^X_{\sigma(1)}[\bar{\Theta}_m],...,\iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)|]_\rho\Delta \downarrow [|E[\bar{\Theta}_n](C_2,a_{\sigma(1)},...,a_{\sigma(n)})(k)|]_\rho\Delta$

$\square$

Lemme 3.54. *Si le jugement* $\Gamma \vdash$ ***valid*** *est dérivable, alors* $\lambda x.(x,\lambda\delta.*) \in [|\Gamma|]\Gamma$.

Démonstration. Par induction sur la structure de Γ.

□

Corollaire 3.55.

- *Si* $\Gamma \vdash A$**kind** *est dérivable, alors il existe une sorte* B *telle que* $\Gamma \vdash^S A \xrightarrow{nf} B$.

- *Si* $\Gamma \vdash M : A$ *est dérivable, alors il existe une sorte* B *et un terme* P *tels que* $\Gamma \vdash^S A \xrightarrow{nf} B$ *et* $\Gamma \vdash^S M \xrightarrow{nf} P : B$.

- *Si* $\Gamma \vdash A = B$ *est dérivable, alors il existe une sorte* C *telle que* $\Gamma \vdash^S A \downarrow B[C]$.

- *Si* $\Gamma \vdash M = N : A$ *est dérivable, alors il existe une sorte* B *et un terme* P *tels que* $\Gamma \vdash^S A \xrightarrow{nf} B$ *et* $\Gamma \vdash^S M \downarrow N[P] : B$.

Démonstration. Considérons le cas du jugement $\Gamma \vdash A$**kind**. Grâce au lemme 3.54, on sait qu'il existe une valuation de Γ vers Γ. On sait également, grâce au lemme 3.53 (Correction), que $[|A|]_\rho\Delta$ est définie. En employant le lemme 3.51, on sait qu'il existe une sorte B telle que le jugement $\Gamma \vdash^S A \xrightarrow{nf} B$ soit dérivable.

□

Les corollaires suivants découlent principalement du lemme 3.53 (Correction).

Corollaire 3.56 (Unicité des types)**.** *Si les jugements* $\Gamma \vdash M : A$ *et* $\Gamma \vdash M : B$ *sont dérivables, alors on a* $\Gamma \vdash A = B$.

Corollaire 3.57 (Normalisation forte)**.** *Si le jugement* $\Gamma \vdash M : A$ *est dérivable, alors* M *est fortement normalisant.*

Corollaire 3.58 (Préservation du type)**.** *Si le jugement* $\Gamma \vdash M : A$ *est dérivable et si* $M \triangleright^* N$*, alors le jugement* $\Gamma \vdash N : A$ *est dérivable.*

Corollaire 3.59 (Confluence)**.** *Si le jugement* $\Gamma \vdash M = N : A$ *est dérivable, alors il existe un terme* P *tel que* $M \triangleright^* P$ *et* $N \triangleright^* P$*.*

Nous venons de démontrer que le système UTT^r vérifie les propriétés de normalisation forte, de confluence et de préservation du type.

Chapitre 4

Sous-typage coercitif

Dans ce chapitre, nous nous situons dans le cadre d'un système de sous-typage coercitif développé pour le "Logical Framework" LF [Luo97c, JLS98, SL00], les coercions y sont vues comme un mécanisme d'abréviation. Le rôle d'un terme de coercion c est illustré par l'utilisation de la règle coercitive suivante :

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k_0 : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_0) = f(c(k_0)) : [c(k_0)/x]K'} \text{ (CD)}$$

Cette règle stipule que si f est une fonction ayant comme domaine K, k_0 un objet de K_0, et c une coercion de K_0 vers K, alors $f(k_0)$ est bien typé et intensionnellement égal à $f(c(k_0))$. Intuitivement, on peut voir f comme un contexte qui requiert un objet de K, alors l'argument k_0 dans le contexte f représente son image par la coercion, $c(k_0)$. Par conséquent, on peut utiliser $f(k_0)$ comme une abréviation de $f(c(k_0))$.

Dans le système formel du sous-typage coercitif, on distingue les coercions de base et les coercions dérivées. Le système des coercions basiques est ouvert, dans le sens où, de nouvelles coercions peuvent être déclarées (par exemple par l'utilisateur). Les nouveaux termes de coercions spécifiés en tant que coercions de base n'ont pas besoin de vérifier des propriétés particulières (ils peuvent être n'importe quelles applications), par contre l'ensemble des coercions de base doit être cohérent. Le fait que les coercions entre les mêmes types soient égales (condition cohérence) est lié à la conservativité de la théorie enrichie du sous-typage coercitif par rapport à la théorie de départ. Dans ce chapitre, nous étudions les propriétés du système UTT^r (système qui intègre une réduction non standard sur les fonctions entre types finis) enrichi du sous-typage coercitif, basées sur celles de l'ensemble des coercions de base (comme la cohérence).

Nous nous concentrons sur les questions du processus de completion des coercions et de la conservativité. Le résultat de conservativité dit intuitivement que, tout jugement dérivable dans le système UTT^r enrichi du sous-typage coercitif, dont la dérivation ne contient pas de règle coercitive, est dérivable dans le système UTT^r. De plus, pour chaque dérivation intégrant des règles coercitives, il est possible d'insérer correctement les termes de coercion manquants afin d'obtenir une dérivation de UTT^r. Aussi, nous démontrons dans ce chapitre la conservativité de UTT^r enrichi du sous-typage coercitif, ce qui justifie que ce concept de sous-typage est vraiment un mécanisme d'abréviation.

La preuve de conservativité est difficile, en partie à cause de la complexité du système de types dépendants que nous étudions. La difficulté vient surtout du fait qu'il faut considérer un algorithme de complétion des coercions basé sur la structure des dérivations, plutôt que sur les jugements ou la syntaxe des termes. Le théorème principal stipulant que la fonction de complétion des coercions est totale, montre que toute dérivation de UTT^r enrichi du sous-typage coercitif, d'un jugement qui n'est pas un jugement de sous-typage, peut être transformée en une dérivation équivalente de UTT^r. Ce résultat justifie non seulement le mécanisme d'abréviation de ce sous-typage mais fournit une justification et des bases intéressantes pour une implémentaion des coercions dans des systèmes d'assistance à la preuve. La preuve de conservativité consiste en trois parties majeures :

- des lemmes sur les propriétés métathéoriques générales de UTT^r enrichi du sous-typage coercitif ;

- l'élimination de la règle de transitivité pour les sous-sortes dans le calcul comprenant les règles de sous-typage, les règles pour les sous-sortes, mais sans les règles de définition et d'application coercitive ;

- La preuve que la fonction de complétion des coercions qui transforme toute dérivation de UTT^r enrichi du sous-typage coercitif en une dérivation dans le calcul sans les règles de définition et d'application coercitive, est totale.

Le résultat principal est que la cohérence de l'ensemble des coercions de bases implique la conservativité. Dans la section 4.1, nous présentons le système formel du sous-typage coercitif. C'est dans la section 4.2 que nous réalisons la preuve de conservativité. Nous commençons par énoncer les principales difficultés inhérentes à la réalisation de la preuve et justifiant la méthode employée que nous décrivons dans la section 4.2.1. Puis, après avoir décrit le plan de la preuve dans la section 4.2.3, nous démontrons tous les lemmes nécessaires à la preuve concernant les propriétés métathéoriques générales de UTT^r enrichi du sous-typage coercitif, incluant l'élimination de la règle de transitivité pour les sous-sortes. Dans la section 4.2.4, nous définissons l'algorithme de complétion des coercions. Pour conclure, dans la section 4.2.5, nous démontrons le fait que UTT^r enrichi du sous-typage coercitif, est une extension conservative de UTT^r comme une conséquence directe du fait que la fonction de complétion de coercion est totale.

4.1 Présentation : Le système formel

L'extension de toute théorie T spécifiée dans LF, avec le sous-typage coercitif, est définie par rapport à un ensemble R de triplets (un type, un sous-type et un terme de coercion qui établit la correspondance entre les éléments du type et ceux de son sous-type). Un de nos principaux objectifs, est d'investir les propriétés générales de toute théorie spécifiée dans LF étendue avec le sous-typage coercitif, notée $T[R]$, ainsi ses relations avec la théorie originale T. Dans cette étude, nous nous consacrons au calcul UTT^r enrichi du sous-typage coercitif, noté $UTT^r[R]$. Il faut également noter, que les résultats obtenus pour $UTT^r[R]$ sont facilement transférables à $UTT[R]$ en raison de la similarité des démonstrations.

4.1.1 Formes de jugements

Avec l'ajout du sous-typage coercitif, de nouvelles formes de jugement apparaissent dans le langage.

- $\Gamma \vdash A <_c B$: **Type** signifie que A est un sous-type de B par le biais de la coercion c.

- $\Gamma \vdash K <_c K'$ signifie que K est une sous-sorte de K' par le biais de la coercion c.

L'étiquette c est appelée terme de coercion. Il sera prouvé dans la suite de ce document que pour tout jugement dérivable $\Gamma \vdash K <_c K'$, le jugement $\Gamma \vdash c : (K)K'$ est également dérivable.

4.1.2 Sous-typage

Nous considérons d'abord un système intermédiaire $UTT^r[R]_o$, qui définit comment les relations de sous-typage sont établies. $UTT^r[R]_o$ est obtenue à partir de UTT^r, en ajoutant les jugements de sous-typage de la forme $\Gamma \vdash A <_c B$: **Type**, ainsi que les règles d'inférence suivantes.

- **Règles basiques de sous-typages**.

  Dans ce document, les règles basiques de sous-typages correspondent à un ensemble R de coercions de base.

  $$\frac{\Gamma \vdash A : \textbf{Type} \quad \Gamma \vdash B : \textbf{Type} \qquad \Gamma \vdash c : (El(A))El(B) \quad (A, B, c) \in R}{\Gamma \vdash A <_c B : \textbf{Type}} \text{ (ST.1)}$$

  A, B et c ne contiennent pas de variable libre.

- **Règles de transitivité pour le sous-typage**.

  $$\frac{\Gamma \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash A = A' : \textbf{Type}}{\Gamma \vdash A' <_c B : \textbf{Type}} \text{ (ST.2)}$$

  $$\frac{\Gamma \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash B = B' : \textbf{Type}}{\Gamma \vdash A <_c B' : \textbf{Type}} \text{ (ST.3)}$$

  $$\frac{\Gamma \vdash A <_{c_1} B : \textbf{Type} \qquad \Gamma \vdash B <_{c_2} C : \textbf{Type}}{\Gamma \vdash A <_{[x:El(A)]c_2(c_1(x))} C : \textbf{Type}} \text{ (ST.4)}$$

- **Règles de congruence pour le sous-typage**.

  $$\frac{\Gamma \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash c = c' : (El(A))El(B)}{\Gamma \vdash A <_{c'} B : \textbf{Type}} \text{ (ST.5)}$$

- **Règle de substitution pour le sous-typage**.

  $$\frac{\Gamma, x : K, \Gamma' \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]A <_{[k/x]c} [k/x]B : \textbf{Type}} \text{ (ST.6)}$$

4.1.3 Condition de cohérence

Définition 4.1 (Condition de cohérence)**.** *On dit que R est cohérent si $UTT^r[R]_o$ vérifie les propriétés suivantes :*

*a) le jugement $\Gamma \vdash A <_c A$: **Type** n'est pas dérivable, quelque soit le contexte Γ, le type A et le terme c ;*

*b) si les jugements $\Gamma \vdash A <_c B$: **Type** et $\Gamma \vdash A <_{c'} B$: **Type** sont dérivables, alors on a $\Gamma \vdash c = c' : (El(A))El(B)$.*

Dans ce papier, nous présumons que R est cohérent.

4.1.4 Sous-sortes

Nous considérons maintenant un nouveau système intermédiaire $UTT^r[R]_{ok}$, obtenu à partir de $UTT^r[R]_o$, par ajout de nouveaux jugements de la forme $\Gamma \vdash K <_c K'$ et des règles d'inférence suivantes.

- **Règles basiques**

$$\frac{\Gamma \vdash A <_c B : \textbf{Type}}{\Gamma \vdash El(A) <_c El(B) : \textbf{Type}} \text{ (SK.1)}$$

- **Sous-sortes et produit dépendant**

$$\frac{\Gamma \vdash K_1' <_{c_1} K_1 \qquad \Gamma, x' : K_1' \vdash [c_1(x')/x]K_2 = K_2' \qquad \Gamma, x : K_1 \vdash K_2 \textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K_1')K_2'} \text{ (SK.2)}$$

avec $c \equiv [f : (x : K_1)K_2][x' : K_1']f(c_1(x'))$;

$$\frac{\Gamma \vdash K_1' = K_1 \qquad \Gamma, x' : K_1' \vdash K_2 <_{c_2} K_2' \qquad \Gamma, x : K_1 \vdash K_2 \textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K_1')K_2'} \text{ (SK.3)}$$

avec $c \equiv [f : (x : K_1)K_2][x' : K_1']c_2(f(x))$;

$$\frac{\Gamma \vdash K_1' <_{c_1} K_1 \qquad \Gamma, x' : K_1' \vdash [c_1(x')/x]K_2 <_{c_2} K_2' \qquad \Gamma, x : K_1 \vdash K_2\mathbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K_1')K_2'} \text{(SK.4)}$$

avec $c \equiv [f : (x : K_1)K_2][x' : K_1']c_2(f(c_1(x')))$;

- **Transitivité pour sous-sortes**

$$\frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_2 = K_2'}{\Gamma \vdash K_1 <_c K_2'} \text{(SK.5)} \qquad \frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_1 = K_1'}{\Gamma \vdash K_1' <_c K_2} \text{(SK.6)}$$

$$\frac{\Gamma \vdash K <_{c_1} K' \qquad \Gamma \vdash K' <_{c_2} K''}{\Gamma \vdash K <_{[x:K]c_2(c_1(x))} K''} \text{(SK.7)}$$

- **Congruence pour sous-sortes**

$$\frac{\Gamma \vdash K <_c K' \qquad \Gamma \vdash c = c' : (K)K'}{\Gamma \vdash K <_{c'} K'} \text{(SK.8)}$$

- **Substitution pour sous-sortes**

$$\frac{\Gamma, x : K, \Gamma' \vdash K_1 <_c K_2 \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]K_1 <_{[k/x]c} [k/x]K_2} \text{(SK.9)}$$

4.1.5 Règles coercitives

L'extension de UTT^r avec le sous-typage et les coercions est le système $UTT^r[R]$. Ce système est obtenu à partir du système intermédiaire $UTT^r[R]_{ok}$, auquel on rajoute les règles d'inférence qui vont suivre. Ce sont ces règles qui établissent la connexion entre UTT^r et son extension avec la notion de sous-typage. En effet, seules ces règles ont pour prémisses des jugements de la forme $\Gamma \vdash K <_c K'$ (spécifique aux jugements de sous-typage et de sous-sortes) et pour conclusion des jugements dont la forme peut être retrouvée dans UTT^r.

- **Règles d'application coercitive**

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k) : [c(k)/x]K'} \text{(CA.1)}$$

$$\frac{\Gamma \vdash f = f' : (x : K)K' \qquad \Gamma \vdash k_1 = k_2 : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_1) = f'(k_2) : [c(k_1)/x]K'} \text{(CA.2)}$$

- **Règle de définition coercitive**

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k_0 : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_0) = f(c(k_0)) : [c(k_0)/x]K'} \text{(CD)}$$

4.2 Preuve de conservativité

4.2.1 Principaux problèmes, méthode et résultats

Commençons par énoncer les principales difficultés inhérentes à la réalisation de la preuve et justifiant la méthode employée.

4.2.1.1 Completion des coercions

La confiance en les coercions comme mécanisme d'abréviation repose sur la possibilité d'insérer de manière uniforme toutes les coercions qui ont été omises lors de l'application des règles CA.1, CA.2 et CD. L'idée principale est la suivante : si nous avons un jugement dont les termes et les sortes composant ce jugement sont dérivables dans UTT^r, ces derniers ne seront pas modifiés par l'algorithme de complétion de coercion et leur $UTT^r[R]$-dérivation deviendra une UTT^r-dérivation.

Exemple 4.2. *Considérons une inférence faisant appel à la règle d'application coercive ($CA.1$).*

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k) : [c(k)/x]K'} \textit{(CA.1)}$$

Nous présumons qu'il existe des UTT^r-dérivations des jugements $\Gamma \vdash f : (x : K)K'$ et $\Gamma \vdash k : K_0$, ainsi qu'une $UTT^r[R]$-dérivation de $\Gamma \vdash K_0 <_c K$. A partir de la $UTT^r[R]$-dérivation de $\Gamma \vdash K_0 <_c K$, il est possible d'obtenir une UTT^r-dérivation du jugement $\Gamma \vdash c : (K_0)K$. Nous pouvons maintenant obtenir une UTT^r-dérivation du jugement $\Gamma \vdash f(k) : [c(k)/x]K'$ de la manière suivante :

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \dfrac{\Gamma \vdash c : (K_0)K \qquad \Gamma \vdash k : K_0}{\Gamma \vdash c(k) : K} \textit{(5.5)}}{\Gamma \vdash f(c(k)) : [c(k)/x]K'} \textit{(5.5)}$$

Les règles $(CA.2)$ et (CD) peuvent être modifiées de la même manière.

Cette construction nous suggère comment on pourrait définir une transformation appelée Ψ, pour la dérivation complète. On partirait du début de la dérivation vers la fin, en remplaçant les jugements de sous-sortes des prémisses des règles $(CA.1)$,$(CA.2)$ et (CD), par la dérivation de leurs termes de coercion et en modifiant les règles de manière adéquate. Le résultat attendu est une UTT^r-dérivation pour les dérivations qui ne contiennent pas de jugement de sous-typage ou de sous-sorte, ou alors respectivement une dérivation dans les extensions conservatives UTT^r, en l'occurrence $UTT^r[R]_o$ et $UTT^r[R]_{ok}$. Cette idée de construction fonctionnera correctement, seulement si on peut garantir une correspondance entre les prémisses de chaque règle ou au moins qu'elles soient égales dans UTT^r. Il faut remarquer qu'un même terme ou sorte peut être modifié de différentes manières selon la dérivation, car différents termes de coercion peuvent être insérés. Cela représente un des cas où l'on constate l'importance des conditions de cohérence. Prenons l'exemple de la $UTT^r[R]$-dérivation suivante :

$$\frac{\Gamma \overset{d_1}{\vdash} f : (x : K_1)K_2 \qquad \Gamma \overset{d_2}{\vdash} k : K_1}{\Gamma \vdash f(k) : [k/x]K_2} \text{(5.5)}$$

Considérons que les $UTT^r[R]$-dérivations des prémisses d_1 et d_2 deviennent, après transformation, les dérivations $\Psi(d_1)$ et $\Psi(d_2)$ dans UTT^r des jugements $\Gamma' \vdash f' : (x : K_1')K_2'$ et $\Gamma'' \overset{d_2}{\vdash} k'' : K_1''$ respectivement. Dans ce cas, les sortes correspondantes dans Γ' et Γ'', devraient être égales dans UTT^r et les mêmes pour K_1' et K_1''. Si toutes ces sortes sont égales dans UTT^r, il existe donc un moyen d'insérer les UTT^r-dérivations de ces égalités.

4.2.1.2 Le problème de conservativité

Nous pouvons dire que le problème de conservativité est un des principaux problèmes, quand nous considérons le sous-typage comme un mécanisme d'abréviation. L'exemple suivant va aider à mieux comprendre son importance ainsi que sa connexion avec les conditions de cohérence.

Exemple 4.3. *Nous affirmons n'avoir aucune connaissance concernant les conditions de cohérence.*

$$\frac{\Gamma \vdash A : \textbf{\textit{Type}} \quad \Gamma \vdash B : \textbf{\textit{Type}} \qquad \Gamma \vdash c_1 : (El(A))El(B) \quad (A, B, c) \in R}{\Gamma \vdash A <_{c_1} B : \textbf{\textit{Type}}} \; \textit{(ST.1)}$$

Considérons qu'il existe un autre terme basique de coercion $\Gamma \vdash c_2 : (El(A))El(B)$ *tel que* $c_1 = c_2$ *n'est pas derivable dans* UTT^r.
En utilisant la règle CD*, il est possible de dériver les jugements suivants :*
$\Gamma, x : El(A) \vdash [y : El(B)]y(c_1(x)) = ([y : El(B)]y)x : El(B)$ *et* $\Gamma, x : El(A) \vdash [y : El(B)]y(c_2(x)) = ([y : El(B)]y)x : El(B)$.
De plus, en appliquant la β*-réduction (règle 5.7), il est possible d'obtenir :* $\Gamma, x : El(A) \vdash c_1(x) = ([y : El(B)]y)x : El(B)$ *et* $\Gamma, x : El(A) \vdash c_2(x) = ([y : El(B)]y)x : El(B)$.
Par la symétrie et la transitivité, on peut avoir $\Gamma, x : El(A) \vdash c_1(x) = c_2(x) : El(B)$*, puis dériver* $\Gamma \vdash [x : El(A)]c_1(x) = [x : El(A)]c_2(x) : (EL(A))El(B)$.
En appliquant la η*-réduction (règle 5.8), le jugement* $\Gamma \vdash c_1(x) = c_2(x) : El(B)$ *peut être dérivé. Ce qui mène à une contradiction.*
Comme le montre cet exemple, les règles de coercions peuvent générer de nouvelles égalités, ce qui est en désaccord avec leur utilisation comme un mécanisme d'abréviation. Il est difficile de prévoir quelles peuvent être les autres conséquences de ces nouvelles égalités.
Les conditions de cohérence sont nécessaires afin d'éviter des situations comme celles rencontrées dans l'exemple ci-dessus.

4.2.1.3 Propriétés méta-théoriques de $UTT^r[R]$ et ses sous-systèmes

Un système relativement riche comme $UTT^r[R]$, contient plusieurs groupes de règles d'inférences qui interagissent entre elles de manière relativement complexe. Pour que ce système soit un instrument efficace, nous devons d'abord étudier ces propriétés basiques, ce qui inclut plusieurs résultats qui peuvent être vus comme une forme d'élimination partielle de coupures, par exemple l'élimination des règles de substitution, de transitivité pour les sous-sortes ou d'affaiblissement.

Un autre groupe de résultats importants, concerne les jugements présupposés et leurs dérivations. Par exemple la dérivabilité d'un jugement de la forme

$$x_1 : Q_1, \dots, x_n : Q_n \vdash K_1 = K_2$$

présuppose la dérivabilité des jugements :
$x_1 : Q_1, \dots, x_{i-1} : Q_{i-1} \vdash Q_i$**kind**, $\quad x_1 : Q_1, \dots, x_i : Q_i \vdash$ **valid** $\quad 1 \leq i \leq n$,
et $x_1 : Q_1, \dots, x_n : Q_n \vdash K_j$**kind** ($j = 1, 2$).
Comment peut-on obtenir ces jugements présupposés et quel est le lien entre la dérivation d'un jugement présupposé et la dérivation principale ?

Ni le problème d'élimination de règles, ni celui des jugements présupposés n'admet de solutions directes. Comme illustration des difficultés possibles, nous verrons quelques exemples. Ces exemples aideront à comprendre l'ordre dans lequel les lemmes peuvent être organisés pour démontrer le résultat principal.

Exemple 4.4. *Essayons d'éliminer la règle de substitution 6.7 par une induction sur la taille de la dérivation.*

$$\dfrac{\dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} k : K'' \qquad \Gamma, x : K, \Gamma' \overset{d_2}{\vdash} K'' = K'}{\Gamma, x : K, \Gamma' \vdash k : K'}\,\textit{(3.1)} \qquad \Gamma \overset{d_3}{\vdash} k_1 = k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]k = [k_2/x]k : [k_1/x]K'}\,\textit{(6.7)}$$

Il semble naturel d'appliquer la règle adéquate de substitution au jugement final de d_1, puis de l'éliminer par hypothèse d'induction, ce qui nous donne la dérivation :

$$\dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} k : K'' \qquad \Gamma \overset{d_3}{\vdash} k_1 = k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]k = [k_2/x]k : [k_1/x]K''}\,\textit{(6.7)}$$

Ensuite il faut remplacer la sorte $[k_1/x]K''$ par $[k_1/x]K'$. Pour cela il nous faudra obtenir la dérivation suivante :

$$\dfrac{\Gamma, x : K, \Gamma' \overset{d_2}{\vdash} K'' = K' \qquad \Gamma \overset{h}{\vdash} k_1 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]K'' = [k_1/x]K'}\,\textit{(6.4)}$$

Enfin, en appliquant la règle 3.2 aux deux jugements dérivés précédemment nous aurions :

$$\frac{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]k' = [k_2/x]k' : [k_1/x]K'' \qquad \Gamma, [k_1/x]\Gamma' \vdash [k_1/x]K'' = [k_1/x]K'}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]k = [k_2/x]k : [k_1/x]K'} \textit{(3.2)}$$

Le problème est qu'il nous manque la prémisse $\Gamma \overset{h}{\vdash} k_1 : K$. *Nous pourrions l'obtenir à partir de* $\Gamma \overset{d_3}{\vdash} k_1 = k_2 : K$ *comme jugement présupposé.*

En même temps, nous pouvons éprouver quelques difficultés à obtenir des jugements présupposés. Considérons par exemple une dérivation se terminant par :

$$\frac{\Gamma \vdash f = f' : (x : K)K' \qquad \Gamma \vdash k_1 = k_2 : K}{\Gamma \vdash f(k_1) = f'(k_2) : [k_1/x]K'} \text{(5.6)}$$

Nous voulons obtenir le jugement $f'(k_2) : [k_1/x]K'$. En procédant par induction sur les prémisses on obtient les jugements $\Gamma \vdash f' : (x : K)K'$ et $\Gamma \vdash k_2 : K$. Si on applique la règle 5.5, on a $f'(k_2) : [k_2/x]K'$ or la sorte devrait être $[k_1/x]K'$. Pour obtenir le jugement désiré, nous aurons sans doute besoin de faire appel aux substitutions, or nous rappelons que nous cherchons à les éliminer également.
Considérons un autre cas, une dérivation se terminant par :

$$\frac{\Gamma, x : K_1 \vdash K_1' = K_2' \qquad \Gamma \vdash K_1 = K_2}{\Gamma \vdash (x : K_1)K_1' = (x : K_2)K_2'} \text{(5.2)}\,,$$

où on cherche des dérivations des deux jugements $\Gamma \vdash (x : K_1)K_1'$**Kind** et $\Gamma \vdash (x : K_2)K_2'$**Kind**. En appliquant l'hypothèse d'induction à la prémisse gauche, on obtient les jugements $\Gamma, x : K_1 \vdash K_1'$**Kind** et $\Gamma, x : K_1 \vdash K_2'$**Kind**. Pour le premier jugement, il suffit d'appliquer la règle 5.1 pour avoir $\Gamma \vdash (x : K_1)K_1'$**Kind**. Par contre, appliquer la règle 5.1 au second jugement nous donne comme résultat $\Gamma \vdash (x : K_1)K_2'$**Kind**. La sorte K_1 devrait être changée en K_2. Pour sauver la situation il faudrait appliquer la règle 3.3 comme suit :

$$\frac{\Gamma, x : K_1 \vdash K_2'\textbf{Kind} \qquad \Gamma \vdash K_1 = K_2}{\Gamma, x : K_2 \vdash K_2'\textbf{Kind}} \text{(3.3)}$$

La règle 3.3 fait partie des règles admissibles que nous souhaitons éliminer.

Pour résoudre ces problèmes, nous commençons par considérer certaines règles dans une formulation affaiblie et moins problématique. A l'aide de ces règles, nous prouvons une variante du théorème d'extraction de jugements présupposés et par la suite, nous considérons le cas des règles qui pose les principales difficultés.

Tous ces résultats sont présentés sous la forme de transformations algorithmiques de dérivations. Par exemple toutes les dérivations utilisant les règles de substitution sont transformées en des dérivations sans présence de règles de substitution, ou encore, de toute dérivation d'un jugement d'égalité, on peut extraire une dérivation du typage d'un des membres de cette égalité, etc.

Avec cette approche, nous conservons plus d'informations sur les dérivations que dans les approches qui utilisent des règles admissibles. La présence de règles admissibles au sein d'une dérivation ne nous fournit aucune information sur la relation entre les dérivations avec ou sans ces règles.

Comme nous avons pu le remarquer précédemment, l'opération de complétion des coercions dépend des dérivations. Ce fait est d'une importance cruciale dans l'étude des propriétés de $UTT^{r}[R]$. L'admissibilité est trop faible pour pouvoir dire quoi que ce soit sur le lien entre la complétion de coercions des dérivations avec ou sans les règles admissibles, ou encore le lien entre les dérivations de certains jugements et les jugements présupposés qu'on peut en extraire. Ce lien peut être établi plus facilement en présence de procédure explicite d'élimination et d'extraction.

4.2.2 Plan de la preuve

1. Dans la section 4.2.3.3, on retrouve les principaux résultats suivants :

   - Les lemmes d'élimination de la règle d'affaiblissement (wkn), d'élimination de la règle de retypage de contexte (3.3), ainsi que ceux d'élimination des substitutions. La règle de retypage et celle de substitution sont considérées dans leurs formulations affaiblies.

   - Les lemmes concernant les jugements présupposés et l'extraction de leurs dérivations à partir de la dérivation du jugement principal. Certaines règles sont également considérées dans leurs formulations affaiblies.

   - L'équivalence entre les règles dans leurs formulations affaiblies et leurs formulations ordinaires est effectuée à l'aide du lemme 4.20.

L'ordre dans lequel apparaissent les lemmes est très important. Cela nous évite de prouver l'ensemble des résultats par une seule et unique induction fastidieuse et très compliquée.

2. Dans la section 4.2.3.4, nous prouvons à l'aide des lemmes précédents que la transitivité pour les sous-sortes peut être éliminée dans la théorie $UTT^r[R]_{ok}$. Pour chaque dérivation d du jugement $\Gamma \vdash K <_c K'$ qui ne contient aucune des règles coercitives $(CA.1)$, $(CA.2)$ et (CD), on peut construire une autre dérivation d' du jugement $\Gamma \vdash K <_{c'} K'$ où $\Gamma \vdash c = c' : (K)K'$ est dérivable dans UTT^r. L'élimination des règles de transitivité pour les sous-sortes permet de montrer que les conditions de cohérence sont vérifiées pour les jugements de sous-sorte, si leurs dérivations ne contiennent pas de règles coercitives et en admettant que les conditions de cohérence soient vérifiées pour le sous-typage.

3. Dans la section 4.2.4.1, nous définissons l'opérateur Ψ de complétion de coercions. A ce niveau, nous ne sommes pas encore en mesure de garantir que Ψ est définie pour toutes les dérivations de $UTT^r[R]$.
4. Dans la section 4.2.4.2, nous montrons la relation qu'il existe entre Ψ et les procédures d'élimination et d'extraction décrites dans la section 4.2.3.3. Pour être plus précis, si $\Psi(d)$ est définie, et si la dérivation d' est obtenue à partir de d par les procédures d'élimination et d'extraction citées plus haut, alors $\Psi(d')$ est également définie, de plus les jugements finaux de $\Psi(d)$ et $\Psi(d')$ sont égaux dans UTT^r.
5. Dans la section 4.2.4.3, on prouve que Ψ est définie pour toutes les dérivations, si les conditions de cohérence sont satisfaites.
6. Pour finir, la section 4.2.5 est consacrée à montrer que le résultat de conservativité est la conséquence du fait que la transformation Ψ est totale.

4.2.3 Propriétés méta-théoriques

Nous explorons dans cette section, les propriétés méta-théoriques du système $UTT^r[R]$.

4.2.3.1 Sous-systèmes

L'exposant "-" ajouté au nom d'un système signifie que les règles suivantes ont été supprimées :

- la règle d'affaiblissement wkn ;
- la règle de retypage de contexte 3.3 ;
- les règles de substitution de LF 6.1-6.7 ;
- les règles de substitution pour le sous-typage et le subkinding $ST.5$, $SK.9$.

Comme $LF \subset UTT^r \subset UTT^r[R]_o \subset UTT^r[R]_{ok}$, nous avons également $LF^- \subset UTT^{r-} \subset UTT^r[R]_o^- \subset UTT^r[R]_{ok}^-$.

Nous considérons également des variantes de certaines règles ayant des prémisses supplémentaires. Ce sont les règles affaiblies. La variante d'une règle r sera nommée r'.

Retypage de contexte

$$\frac{\Gamma, x : K, \Gamma' \vdash J \qquad \Gamma \vdash K = K' \quad \Gamma \vdash K' \mathbf{kind}}{\Gamma, x : K', \Gamma' \vdash J} \text{ (3.3')}$$

Les règles de substitutions

$$\frac{\Gamma, x : K, \Gamma' \vdash K' \mathbf{kind} \qquad \Gamma \vdash k_1 = k_2 : K \quad \Gamma \vdash k_1 : K \quad \Gamma \vdash k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]K' = [k_2/x]K'} \text{ (6.6')}$$

$$\frac{\Gamma, x : K, \Gamma' \vdash k' : K' \qquad \Gamma \vdash k_1 = k_2 : K \quad \Gamma \vdash k_1 : K \quad \Gamma \vdash k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]k' = [k_2/x]k' : [k_1/x]K'} \text{ (6.7')}$$

Transitivité et congruence pour le sous-typage

$$\frac{\Gamma \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash A = A' : \textbf{Type} \quad \Gamma \vdash A' : \textbf{Type}}{\Gamma \vdash A' <_c B : \textbf{Type}} \text{(ST.2')}$$

$$\frac{\Gamma \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash B = B' : \textbf{Type} \quad \Gamma \vdash B' : \textbf{Type}}{\Gamma \vdash A <_c B' : \textbf{Type}} \text{(ST.3')}$$

Transitivité et congruence pour les sous-sortes

$$\frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_2 = K_2' \quad \Gamma \vdash K_2' \textbf{kind}}{\Gamma \vdash K_1 <_c K_2'} \text{(SK.5')}$$

$$\frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_1 = K_1' \quad \Gamma \vdash K_1' \textbf{kind}}{\Gamma \vdash K_1' <_c K_2} \text{(SK.6')}$$

On dénote par $UTT^r[R]_w$, le système $UTT^r[R]$ dans lequel les règles affaiblies vues plus haut, remplacent celles correspondantes dans leur formulation ordinaire. Le calcul $UTT^r[R]_w^-$ est obtenu de manière similaire à partir de $UTT^r[R]^-$. Nous montrerons par la suite, l'équivalence entre les calculs avec la formulation affaiblie et ceux avec la formulation ordinaire des règles vues plus haut. Il faut bien noter qu'aucun calcul que nous considérerons, ne contient à la fois les règles dans leur formulation affaiblie et ordinaire.

4.2.3.2 Propriétés basiques

Nous présentons ici quelques propriétés basiques de notre système qui nous seront utiles pour réaliser la preuve de conservativité. Nous commençons par introduire quelques définitions, et nous fixons certaines notations.

Définition 4.5 (Constituants d'un jugement)**.**
Nous appelons constituants d'un jugement $\Gamma \vdash J$*, les éléments suivants :*

- *les sortes des variables de* Γ *;*

- *les sortes des variables liées ;*

- *Si $J \equiv K$**kind** ou $J \equiv K_1 = K_2$ alors K, respectivement K_1 et K_2 sont des constituants ;*

- *Si $J \equiv k : K$ ou $J \equiv k_1 = k_2 : K$ alors k et K, respectivement k_1, k_2 et K sont des constituants ;*

- *Si $J \equiv A <_c B :$ **Type** alors c, A, B et **Type** sont des constituants ;*

- *Si $J \equiv K_1 <_c K_2$ alors c,K_1 et K_2 sont des constituants.*

*A, B, c, k, k_1 et k_2 sont appelés des terme-constituants tandis que K, K_1, K_2 et **Type** sont des sorte-constituants de $\Gamma \vdash J$. Ces constituants sont considérés différents dans chacune de leurs occurrences.*

Remarque 4.6 (Notation)**.**
Soit $\Gamma' \equiv x'_1 : K'_1, \ldots, x'_n : K'_n$, et $\Gamma'' \equiv x''_1 : K''_1, \ldots, x''_n : K''_n$, nous admettons l'abréviation $\Gamma \vdash \Gamma' = \Gamma''$ signifiant :
$\Gamma \vdash K'_1 = K''_1, \quad \Gamma, x : K''_1 \vdash K'_2 = K''_2, \quad \ldots, \quad \Gamma, x : K''_1, \ldots, x_{n-1} : K''_{n-1} \vdash K'_n = K''_n$.

Nous considérons également comme abréviation l'égalité entre des jugements de la même forme. Cela inclut l'égalité entre les contextes introduits précédemment ainsi que l'égalité entre les constituants se trouvant du côté droit du signe "$\vdash$" des jugements. Il faut faire attention à l'ordre dans lequel les égalités sont considérées. Par exemple l'expression $\Gamma \vdash k : K = \Gamma' \vdash k' : K'$ est une abréviation pour $\vdash \Gamma = \Gamma'$, $\Gamma \vdash K = K'$ et $\Gamma \vdash k = k' : K$.

Pour les règles d'inférences nous utiliserons l'abréviation

$$\frac{\Gamma, \Gamma' \vdash J \qquad \Gamma \vdash \Gamma' = \Gamma''}{\Gamma, \Gamma'' \vdash J} \text{ (3.3)}$$

pour représenter l'inférence suivante :

$$\dfrac{\dfrac{\dfrac{\Gamma, x_1 : K'_1, \ldots, x_n : K'_n \vdash J \qquad \Gamma \vdash K'_1 = K''_1}{\Gamma, x_1 : K''_1, \ldots, x_n : K'_n \vdash J}\,\textit{(3.3)}}{\cdots\cdots\cdots} \qquad \dfrac{}{\Gamma, x_1 : K''_1, \ldots, x_{n-1} : K''_{n-1} \vdash K'_n = K''_n}}{\Gamma, x_1 : K''_1, \ldots, x_n : K''_n \vdash J}\,\textit{(3.3)}$$

(... signifie une série appropriée de règle de retypage 3.3).
Une abréviation similaire sera utilisée pour une série de règles de retypage utilisant la formulation affaiblie $3.3'$.

Si on considère deux dérivations d *et* d' *, la notation* $\mathrm{d} \sim \mathrm{d}'$ *signifie que les jugements finaux de* d *et* d' *sont égaux dans le sens décrit précédemment.*

Passons maintenant aux propriétés basiques de notre système.

Lemme 4.7. $UTT^r[R]_o$ *est une extension conservative de* UTT^r*, c'est-à-dire que, si l'expression* J *n'est pas de la forme* $A <_c B$: ***Type****, alors le jugement* $\Gamma \vdash J$ *est dérivable dans* UTT^r *si et seulement si* $\Gamma \vdash J$ *est dérivable dans* $UTT^r[R]_o$*.*

Lemme 4.8. *Les conditions de cohérence sont vérifiées pour le système* $UTT^r[R]$*.*

Lemme 4.9. $UTT^r[R]_{ok}$ *est une extension conservative de* $UTT^r[R]_o$*, c'est-à-dire que, si l'expression* J *n'est pas de la forme* $K <_c K'$*, alors le jugement* $\Gamma \vdash J$ *est dérivable dans* $UTT^r[R]_o$ *si et seulement si* $\Gamma \vdash J$ *est dérivable dans* $UTT^r[R]_{ok}$*.*

Lemme 4.10.

(a) Si un jugement de la forme $\Gamma \vdash K$***kind*** *est dérivable alors* K *est soit une El-sorte (sorte de la forme El(t)), une sorte-produit(sorte de la forme* $(x : Q)Q'$*) ou la sorte* ***Type****.*

(b) Si un jugement de la forme $\Gamma \vdash K = K'$ *est dérivable alors* K *et* K' *sont simultanément soit des El-sortes, des sorte-produits ou les sortes* ***Type****.*

(c) *Si un jugement de la forme $\Gamma \vdash K <_c K'$ est dérivable alors K et K' sont simultanément soit des El-sortes, des sorte-produits ou les sortes **Type**.*

Démonstration. Preuve par induction sur la taille des dérivations.

(a) **Cas de base**. C'est lorsque l'on rencontre une des règles 4.1, 4.2 ou 5.1.
Pas d'induction. Les règles à considérer sont les règles 4.1, 4.2, 5.1 et 6.2. Evident pour 4.1, 4.2 et 5.1. Pour 6.2, par hypothèse d'induction le lemme est vrai pour K, donc vrai pour $[k/x]K$ car la substitution, selon sa définition ne change pas la structure d'une sorte.

(b) **Cas de base**. C'est lorsque l'on rencontre une des règles 2.1 ou 4.3.
Pas d'induction. Les règles à considérer sont les règles 2.1, 2.2, 2.3, 4.3, 5.2, et 6.4. Evident pour 2.1 et 4.3. Prenons l'exemple de 2.2, par hypothèse d'induction le lemme est vrai pour K et K', c'est ce qu'on souhaite démontrer. Pour 6.4, par hypothèse d'induction le lemme est vrai pour K et K', il l'est donc pour $[k/x]K$ et $[k/x]K'$ car la substitution, selon sa définition ne change pas la structure d'une sorte.

(c) **Cas de base**. C'est lorsque l'on rencontre une des règles SK.1 -SK.4.
Pas d'induction. Les règles à considérer sont toutes les règles pour les sous-sortes. Evident pour SK.1 -SK.4. Exemple pour SK.5, par hypothèse d'induction et grâce au b), le lemme est vrai pour K_1 et K_2', c'est ce qu'on souhaite démontrer. Pour SK.9, par hypothèse d'induction le lemme est vrai pour K_1 et K_2, il l'est donc pour $[k/x]K_1$ et $[k/x]K_2$, car la substitution, selon sa définition ne change pas la structure d'une sorte .

□

Lemme 4.11. *Si un jugement de la forme $\Gamma \vdash J$ est dérivable, alors $FV(J) \subseteq FV(\Gamma)$.*

Démonstration. Par induction sur les dérivations.

□

Lemme 4.12. *Si le jugement $\Gamma, x : K \vdash$ **valid** est dérivable, alors $x \notin FV(\Gamma)$.*

Démonstration. Preuve par induction sur la taille des dérivations.
Cas de base. Quand apparaît la règle 1.2.
Pas d'induction. les règles à considérer sont wkn, 3.3, 6.1. On applique l'hypothèse d'induction aux prémisses.

□

4.2.3.3 Eliminations et jugements présupposés

Nous présentons dans cette section des algorithmes permettant d'éliminer des règles admissibles et de construire des dérivations de jugements présupposés, à partir de la dérivation d'un jugement principal.

Lemme 4.13 (Sous-dérivations)**.** *Nous nous positionnons dans le système $UTT^r[R]^-$.*

(a) Toute dérivation d *du jugement* $\Gamma_1, \Gamma_2 \vdash J$ *contient une sous-dérivation* $V^{\circ}_{\Gamma_1}(d)$ *du jugement* $\Gamma_1 \vdash$ ***valid****.*

(b) Toute dérivation d *du jugement* $\Gamma_1, x : K, \Gamma_2 \vdash J$ *contient une sous-dérivation* $K^{\circ}_{\Gamma_1}(d)$ *du jugement* $\Gamma_1 \vdash K$***Kind****.*

(c) Toute dérivation d *du jugement* $\Gamma \vdash (x : K_1)K_2$ *contient une sous-dérivation* $p\_^{\circ}(d)$ *du jugement* $\Gamma_1, x : K_1 \vdash K_2$***Kind****.*

Démonstration. Par induction sur la taille (le nombre de règles) des dérivations des prémisses gauches. Nous considérons quelques cas à titre d'exemple.

(a) **Cas de base**.

$$\text{d}\equiv \frac{}{\Gamma_1, \Gamma_2 \vdash \textbf{valid}}\ (1.1)$$

avec $\Gamma_2 \equiv \emptyset$. $V^{\circ}_{\Gamma_1}$(d)$\equiv \Gamma_1 \vdash$ **valid**.

Pas d'induction. Considérons la règle (ϑ-eq) :

$$\text{d}\equiv \frac{\Gamma_1, \Gamma_2 \overset{d_1}{\vdash} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma_1, \Gamma_2 \vdash E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m \qquad \Gamma_1, \Gamma_2 \vdash k : F_n}{\Gamma_1, \Gamma_2 \vdash E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k) = E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A}$$

on a $V^{\circ}_{\Gamma_1}$(d)$\equiv \Gamma_1 \overset{d_1}{\vdash}$ **valid**.

(b) Montrons d'abord que toute dérivation d du jugement $\Gamma_1, x : K \vdash$ **valid** contient une sous-dérivation K'_{Γ_1}(d) du jugement $\Gamma_1 \vdash K$**kind**.
Seule la règle (1.2) est à considérer :

$$\mathrm{d}\equiv \frac{\Gamma \overset{d_1}{\vdash} K\textbf{kind}}{\Gamma, x : K \vdash \textbf{valid}} \;(1.2)$$

$K'_{\Gamma_1}(d) \equiv d_1$.

Revenons à $K^{\circ}_{\Gamma_1}$.
Pour toute dérivation d du jugement $\Gamma_1, x : K, \Gamma_2 \vdash J$, on a $d_1 \equiv \Gamma_1, x : K \overset{V^{\circ}_{\Gamma_1}(d)}{\vdash}$ **valid**, de plus $\Gamma_1 \overset{K'_{\Gamma_1}(d_1)}{\vdash} K$**kind**, donc $K^{\circ}_{\Gamma_1}(d) \equiv K'_{\Gamma_1}(V^{\circ}_{\Gamma_1}(d_1))$.

(c) Seule la règle (5.1) est à considérer :

$$\mathrm{d}\equiv \frac{\Gamma, x : K_1 \overset{d_1}{\vdash} K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2\textbf{kind}} \;(5.1)$$

$p\_^{\circ}(d) \equiv d_1$.

□

Nous passons maintenant aux dérivations de $UTT^r[R]^-$ étendues par certaines règles.

Lemme 4.14. *Soit la dérivation* $d \equiv \dfrac{\Gamma_1, \Gamma_2 \overset{d_1}{\vdash} J \qquad \Gamma_1, \Gamma_3 \overset{d_2}{\vdash} \textbf{\textit{valid}}}{\Gamma_1, \Gamma_2, \Gamma_3 \vdash J}$ *(wkn) où les dérivations* d_1 *et* d_2 *ne contiennent pas la règle wkn. Il existe un algorithme* E°_{wkn} *qui transforme* d *en une dérivation du même jugement mais ne contenant pas la règle wkn.*

Démonstration. Preuve par induction sur la taille de d_1.
Cas de base. Le cas de base est le suivant :

$$d \equiv \frac{\dfrac{}{<>\vdash \textbf{valid}}\;(1.1) \qquad \Gamma_3 \overset{d_2}{\vdash} \textbf{valid}}{\Gamma_3 \vdash \textbf{valid}} \;(\text{wkn})$$

Avec $\Gamma_1 = \Gamma_2 = \emptyset$. Dans ce cas $\mathbf{E}^\circ_{wkn}(d) \equiv d_2$.

Pas d'induction. Nous considérons les dernières prémisses de la dérivation d_1 auxquelles nous appliquons la règle (wkn) avec la conclusion de la dérivation d_2. Nous éliminons ensuite (wkn) par hypothèse d'induction. On applique la dernière règle de d_1.

- Considérons le cas suivant comme illustration :

$$d \equiv \cfrac{\cfrac{\Gamma_1, \Gamma_2, y : Q \overset{d_0}{\vdash} Q'\mathbf{kind}}{\Gamma_1, \Gamma_2 \vdash (y : Q)Q'\mathbf{kind}}\,(5.1) \qquad \Gamma_1, \Gamma_3 \overset{d_2}{\vdash} \mathbf{valid}}{\Gamma_1, \Gamma_3, \Gamma_2 \vdash (y : Q)Q'\mathbf{kind}}\,(\text{wkn})$$

Par hypothèse d'induction la dérivation suivante est définie

$$d_0' \equiv \mathbf{E}^\circ_{wkn}\left(\frac{\Gamma_1, \Gamma_2, y : Q \overset{d_0}{\vdash} Q'\mathbf{kind} \qquad \Gamma_1, \Gamma_3 \overset{d_2}{\vdash} \mathbf{valid}}{\Gamma_1, \Gamma_3, \Gamma_2, y : Q \vdash Q'\mathbf{kind}}\,(\text{wkn})\right)$$

on a $\mathbf{E}^\circ_{wkn}(d) \equiv \dfrac{\Gamma_1, \Gamma_3, \Gamma_2, y : Q \overset{d_0'}{\vdash} Q'\mathbf{kind}}{\Gamma_1, \Gamma_3, \Gamma_2 \vdash (y : Q)Q'\mathbf{kind}}\,(5.1)$

- Le seul cas exceptionnel est le suivant :

$$d \equiv \cfrac{\cfrac{\Gamma_0 \overset{d_0}{\vdash} K\mathbf{kind}}{\Gamma_0, x : K \vdash \mathbf{valid}}\,(1.2) \qquad \Gamma_0, x : K, \Gamma_3 \overset{d_2}{\vdash} \mathbf{valid}}{\Gamma_0, x : K, \Gamma_3 \vdash \mathbf{valid}}\,(\text{wkn})$$

Avec $\Gamma_1 = \Gamma_0, x : K$ et $\Gamma_2 = \emptyset$. Dans ce cas $\mathbf{E}^\circ_{wkn}(d) \equiv d_2$.

$\square$

Lemme 4.15. *Soit la dérivation* $d \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} J \qquad \Gamma \overset{d_2}{\vdash} K = K' \qquad \Gamma \overset{d_3}{\vdash} K'\boldsymbol{kind}}{\Gamma, x : K', \Gamma' \vdash J}$ *(3.3')* *où les dérivations* d_1*,* d_2 *et* d_3 *ne contiennent pas la règle* $3.3'$*. Il existe un algorithme* $\boldsymbol{E}^\circ_{rt}$ *qui transforme* d *en une dérivation du même jugement mais ne contenant pas la règle* $3.3'$*.*

Démonstration. Preuve par induction sur la taille de d_1.

Cas de base.

$$\text{d}\equiv \frac{\Gamma, x : K \overset{d_1}{\vdash} \textbf{valid} \qquad \Gamma \overset{d_2}{\vdash} K = K' \quad \Gamma \overset{d_3}{\vdash} K'\textbf{kind}}{\Gamma, x : K' \vdash \textbf{valid}}\ (3.3')$$

$$\mathbf{E}^{\circ}_{rt}(\text{d})\equiv \frac{\Gamma \overset{d_3}{\vdash} K'\textbf{kind}}{\Gamma, x : K' \vdash \textbf{valid}}\ (1.2)$$

Pas d'induction.

- Pour la plupart des cas, on applique la règle $3.3'$ aux prémisses de d_1 que l'on élimine par hypothèse d'induction. Prenons l'exemple suivant comme illustration :

$$\text{d}\equiv \frac{\dfrac{\Gamma, x : K, \Gamma', y : Q \overset{d_0}{\vdash} Q'\textbf{kind}}{\Gamma, x : K, \Gamma' \vdash (y : Q)Q'\textbf{kind}}\ (5.1) \qquad \Gamma \overset{d_2}{\vdash} K = K' \quad \Gamma \overset{d_3}{\vdash} K'\textbf{kind}}{\Gamma, x : K', \Gamma' \vdash (y : Q)Q'\textbf{kind}}\ (3.3')$$

  Par hypothèse d'induction la dérivation suivante est définie

$$d_0' \equiv \mathbf{E}^{\circ}_{rt}(\frac{\Gamma, x : K, \Gamma', y : Q \overset{d_0}{\vdash} Q'\textbf{kind} \qquad \Gamma \overset{d_2}{\vdash} K = K' \quad \Gamma \overset{d_3}{\vdash} K'\textbf{kind}}{\Gamma, x : K', \Gamma', y : Q \vdash Q'\textbf{kind}}\ (3.3'))$$

  on a $\mathbf{E}^{\circ}_{rt}(d) \equiv \dfrac{\Gamma, x : K', \Gamma', y : Q \overset{d_0'}{\vdash} Q'\textbf{kind}}{\Gamma, x : K', \Gamma' \vdash (y : Q)Q'\textbf{kind}}\ (5.1)$

- Considérons le cas particulier suivant :

$$\text{d}\equiv \frac{\dfrac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K}\ (1.3) \qquad \Gamma \overset{d_2}{\vdash} K = K' \quad \Gamma \overset{d_3}{\vdash} K'\textbf{kind}}{\Gamma, x : K', \Gamma' \vdash x : K}\ (3.3')$$

  Par hypothèse d'induction la dérivation suivante est définie

$$d_0' \equiv \mathbf{E}^{\circ}_{rt}(\frac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid} \qquad \Gamma \overset{d_2}{\vdash} K = K' \quad \Gamma \overset{d_3}{\vdash} K'\textbf{kind}}{\Gamma, x : K', \Gamma' \vdash \textbf{valid}}\ (3.3'))$$

Nous prenons $\mathbf{E}^{\circ}_{rt}(d) \equiv$

$$\dfrac{\dfrac{\Gamma, x : K', \Gamma' \overset{d'_0}{\vdash} \textbf{valid}}{\Gamma, x : K', \Gamma' \vdash x : K'}\,(1.3) \qquad \mathbf{E}^{\circ}_{wkn}\Big(\dfrac{\dfrac{\Gamma \overset{d_2}{\vdash} K = K'}{\Gamma \vdash K' = K}\,(2.2) \qquad \Gamma, x : K', \Gamma' \overset{d'_0}{\vdash} \textbf{valid}}{\Gamma, x : K', \Gamma' \vdash K' = K}\,(\text{wkn})\Big)}{\Gamma, x : K', \Gamma' \vdash x : K}\,(3.2)$$

$\square$

Lemme 4.16. *Il existe des algorithmes* $\mathbf{l}^{\circ}_{<}$*,* $\mathbf{r}^{\circ}_{<}$*,* $\mathbf{co}^{\circ}$ *qui extraient d'une* $UTT^{r}[R]^{-}_{w}$*-dérivation* d *d'un jugement de la forme* $\Gamma \vdash A <_c B : \textbf{Type}$*, les dérivations de* $\Gamma \vdash A : \textbf{Type}$*,* $\Gamma \vdash B : \textbf{Type}$ *et* $\Gamma \vdash c : (El(A))El(B)$ *respectivement. Si* d *est une dérivation de* $\Gamma \vdash K <_c K'$ *alors ces algorithmes extraient les dérivations de* $\Gamma \vdash K\textbf{kind}$*,* $\Gamma \vdash K'\textbf{kind}$ *et* $\Gamma \vdash c : (K)K'$*.*

Démonstration. Induction sur la taille des dérivations. Considérons les cas suivants :

- Cas :

$$\text{d}\equiv \dfrac{\Gamma \overset{d_1}{\vdash} A : \textbf{Type} \qquad \Gamma \overset{d_2}{\vdash} B : \textbf{Type} \qquad \Gamma \overset{d_3}{\vdash} c : (El(A))El(B) \qquad (A, c, B) \in R}{\Gamma \vdash A <_c B : \textbf{Type}}\ \text{ST.1}$$

  Ce cas couvre le cas de base ainsi que les pas d'induction. $\mathbf{l}^{\circ}_{<}(d) \equiv d_1$, $\mathbf{r}^{\circ}_{<}(d) \equiv d_2$ et $\mathbf{co}^{\circ}(d) \equiv d_3$.

- Cas :

$$\text{d}\equiv \dfrac{\Gamma \overset{d_1}{\vdash} A <_c B : Type \qquad \Gamma \overset{d_2}{\vdash} A = A' : Type \qquad \Gamma \overset{d_3}{\vdash} A' : Type}{\Gamma \vdash A' <_c B : Type}\ \text{ST.2'}$$

  On a $\mathbf{l}^{\circ}_{<}(d) \equiv d_3$, $\quad \mathbf{r}^{\circ}_{<}(d) \equiv \mathbf{r}^{\circ}_{<}(d_1)$.
  De plus $\mathbf{co}^{\circ}(d)$ est définie de la manière suivante :

$$\dfrac{\dfrac{\mathbf{E}^{\circ}_{wkn}\Big((\text{wkn})\ \dfrac{\dfrac{\dfrac{\Gamma \overset{\mathbf{r}^{\circ}_{<}(d_1)}{\vdash} B : \textbf{Type}}{\Gamma \vdash El(B)\textbf{Kind}}\,(4.2)}{\Gamma \vdash El(B) = El(B)}\,(2.1) \qquad \dfrac{\dfrac{\Gamma \overset{\mathbf{l}^{\circ}_{<}(d_1)}{\vdash} A : \textbf{Type}}{\Gamma \vdash El(A)\textbf{Kind}}\,(4.2)}{\Gamma, x : El(A) \vdash \textbf{valid}}\,(1.2)}{\Gamma, x : El(A) \vdash El(B) = El(B)}\Big) \qquad \dfrac{\Gamma \overset{d_2}{\vdash} A = A' : \textbf{Type}}{\Gamma \vdash El(A) = El(A')}\,(4.3)}{\Gamma \vdash x : (El(A))El(B) = (El(A'))El(B)}\,(5.2) \qquad \Gamma \overset{\mathbf{co}^{\circ}(d_1)}{\vdash} c : (El(A))El(B)}{\Gamma \vdash c : (El(A'))El(B)}\,(3.1)$$

$\square$

Lemme 4.17. *Soit la dérivation* $d \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} J \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]J}$ *(sub), où (sub) représente une des règles 6.1-6.5 ou SK.9. Les sous-dérivations* d_1 *et* d_2 *ne contiennent pas de règle de substitution. Il existe un algorithme* $\mathbf{E}_{sub}$ *qui transforme la dérivation* d *en une dérivation sans les règles représentées par (sub).*

Démonstration. Prenons la dérivation $d \equiv$

$$\dfrac{\dfrac{\Gamma, x : K, \Gamma' \overset{D_1}{\vdash} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma, x : K, \Gamma' \overset{D_2}{\vdash} E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m \qquad \Gamma, x : K, \Gamma' \overset{D_3}{\vdash} q : F_n}{\Gamma, x : K, \Gamma' \vdash E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(q) = E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(q) : A}\,(\vartheta\text{-eq}) \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x](E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(q)) = [k/x](E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(q)) : A}\,(6.5)$$

On a $\mathbf{E}_{sub}(d) \equiv$

$$\dfrac{\Gamma, [k/x]\Gamma' \overset{D'_1}{\vdash} [k/x]E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma, [k/x]\Gamma' \overset{D'_2}{\vdash} [k/x]E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m \qquad \Gamma, [k/x]\Gamma' \overset{D'_3}{\vdash} [k/x]q : F_n}{\Gamma, [k/x]\Gamma' \vdash [k/x]E[\bar{\Theta}_m](C, a_1, ..., a_m)([k/x]E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])([k/x]q) = [k/x]E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})([k/x]q) : A}\,(\vartheta\text{-eq})$$

avec $D'_1 \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{D_1}{\vdash} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A}\,(6.3)$

$D'_2 \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{D_1}{\vdash} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A}\,(6.3)$

et $D'_3 \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{D_1}{\vdash} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A}\,(6.3)$

Notons que
$[k/x]E[\bar{\Theta}_m](C, a_1, ..., a_m)([k/x]E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])([k/x]q) =$
$[k/x](E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(q))$
et que
$[k/x]E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})([k/x]q) = [k/x](E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(q))$

Il existe deux cas spéciaux n'employant pas la même méthode :

- d_1 se termine par (1.2) : $d \equiv \dfrac{\dfrac{\Gamma \overset{d_0}{\vdash} K\,\mathbf{Kind}}{\Gamma, x : K \vdash \mathbf{valid}}\,(1.2) \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma \vdash \mathbf{valid}}\,(6.1)$

nous prenons $\mathbf{E}_{sub}(d) \equiv \Gamma \overset{V^{\circ}_{\Gamma_1}(d_0)}{\vdash}$ **valid** ou $\mathbf{E}_{sub}(d) \equiv \Gamma \overset{V^{\circ}_{\Gamma_1}(d_2)}{\vdash}$ **valid**.

- d_1 se termine par (1.3) :
$$d \equiv \cfrac{\cfrac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K}(1.3) \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash k : K}(6.3)$$

nous avons :

$$\mathbf{E}_{sub}(d) \equiv \mathbf{E}^{\circ}_{wkn}\left(\cfrac{\Gamma \overset{d_2}{\vdash} k : K \qquad \mathbf{E}_{sub}\left(\cfrac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid} \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash \textbf{valid}}(6.1)\right)}{\Gamma, [k/x]\Gamma' \vdash k : K}(\text{wkn})\right)$$

□

Lemme 4.18. *(On considère ici les règles affaiblies). Soit la dérivation*
$$d \equiv \cfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} K' \textbf{\textit{Kind}} \qquad \Gamma \overset{d_2}{\vdash} k_1 = k_2 : K \qquad \Gamma \overset{d_3}{\vdash} k_1 : K \quad \Gamma \overset{d_4}{\vdash} k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]K' = [k_2/x]K'}\textit{(6.6')}$$
ou
$$d \equiv \cfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} k' : K' \qquad \Gamma \overset{d_2}{\vdash} k_1 = k_2 : K \qquad \Gamma \overset{d_3}{\vdash} k_1 : K \quad \Gamma \overset{d_4}{\vdash} k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]k' = [k_2/x]k' : [k_1/x]K'}\textit{(6.7')}$$
*Les dérivations d_1, d_2, d_3, d_4 ne contiennent pas de règle de substitution. Il existe un algorithme **E**$_{eq}$ qui transforme d en une dérivation du même jugement mais ne contenant pas de règle de substitution.*

Démonstration. Soit la dérivation
$$\text{d} \equiv \cfrac{\cfrac{\Gamma, x : K, \Gamma' \vdash \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash \textbf{Type Kind}}(4.1) \qquad \Gamma \overset{d_2}{\vdash} k_1 = k_2 : K \qquad \Gamma \overset{d_3}{\vdash} k_1 : K \quad \Gamma \overset{d_4}{\vdash} k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash \textbf{Type} = \textbf{Type}}(6.6')$$

On a $\mathbf{E}_{eq}(d) \equiv$

$$\mathbf{E}_{sub}(\cfrac{\cfrac{\Gamma, x : K, \Gamma' \vdash \textbf{Type Kind} \qquad \Gamma \overset{d_3}{\vdash} k_1 : K}{\Gamma, [k_1/x]\Gamma' \vdash \textbf{Type Kind}}\ (6.2)}{\Gamma, [k_1/x]\Gamma' \vdash \textbf{Type} = \textbf{Type}}\ (2.1))$$

□

Lemme 4.19. *Il existe un algorithme* $\boldsymbol{E}^{\circ}$*, qui transforme toute* $UTT^r[R]_w$*-dérivation* d *en une* $UTT^r[R]_w^-$*-dérivation.*

Démonstration. A l'aide des algorithmes d'élimination définis précédemment.

□

On considère maintenant des cas plus compliqués de jugements présupposés.

Lemme 4.20. *(Split-lemma). Il existe des algorithmes* $\boldsymbol{l}^{\circ}_{=}$*,* $\boldsymbol{r}^{\circ}_{=}$*,* $l^{\circ}_{=}$*,* $r^{\circ}_{=}$ *et* $\boldsymbol{kd}^{\circ}$ *qui transforment toute* $UTT^r[R]_w^-$*-dérivation* d *de*

1) $\Gamma \vdash K_1 = K_2$ *en deux dérivations* $\boldsymbol{l}^{\circ}_{=}(d)$ *de* $\Gamma \vdash K_1\boldsymbol{kind}$ *et* $\boldsymbol{r}^{\circ}_{=}(d)$ *de* $\Gamma \vdash K_2\boldsymbol{kind}$ *;*

2) $\Gamma \vdash k_1 = k_2 : K$ *en deux dérivations* $l^{\circ}_{=}(d)$ *de* $\Gamma \vdash k_1 : K$ *et* $r^{\circ}_{=}(d)$ *de* $\Gamma \vdash k_2 : K$ *;*

3) $\Gamma \vdash \Sigma : K$ *en une dérivation* $\boldsymbol{kd}^{\circ}(d)$ *de* $\Gamma \vdash K\boldsymbol{kind}$ *(*Σ *denote un terme ou une égalité entre termes).*

Démonstration. Voyons les trois cas simultanément par induction structurelle sur les dérivations.

Cas de base. Ce sont les cas où l'on ne retrouve pas comme prémisse de la dernière règle de d, l'égalité pour 1) et 2) ou un jugement de typage pour 3). Ce qui inclut la règle 2.1 pour 1), la règle 2.4 et les déclarations d'égalité pour 2) et la règle 1.3 et les déclarations de constante pour 3).

1) si $d \equiv \cfrac{\Gamma \overset{d_0}{\vdash} K\textbf{kind}}{\Gamma \vdash K = K}\ (2.1)$; alors $\mathbf{l}^{\circ}_{=}(d) \equiv \mathbf{r}^{\circ}_{=}(d) \equiv d_0$.

2) • si $d \equiv \dfrac{\Gamma \overset{d_0}{\vdash} k : K}{\Gamma \vdash k = k : K}(2.4)$; alors $l^\circ_=(d) \equiv r^\circ_=(d) \equiv d_0$.

• si d se termine par une égalité déclarée, $d \equiv \dfrac{\ldots \Gamma \overset{D_1}{\vdash} k_1 : K \quad \Gamma \overset{D_2}{\vdash} k_2 : K}{\Gamma \vdash k_1 = k_2 : K}$;
alors $l^\circ_=(d) \equiv D_1$ et $r^\circ_=(d) \equiv D_2$.

3) • si $d \equiv \dfrac{\Gamma, x : K, \Gamma'\Gamma \overset{d_0}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K}(1.3)$ alors

$$\textbf{kd}^\circ(d) \equiv \textbf{E}^\circ\Big(\frac{\Gamma \overset{K^\circ_{\Gamma_1}(d_0)}{\vdash} K\textbf{kind} \quad \Gamma, x : K, \Gamma'\Gamma \overset{d_0}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash K\textbf{kind}} \text{wkn}\Big)$$

• Il faut également considérer les cas où d se termine par une déclaration de constante. Par exemple, prenons le cas d'une constante de sorte **Type**, on retrouve facilement le jugement $\Gamma \vdash$ **Type kind**.

Pas d'induction pour le cas 1). Les dérivations peuvent se terminer par les règles suivantes :2.2, 2.3, 4.3, 5.2.

• Cas $d \equiv \dfrac{\Gamma \overset{d_0}{\vdash} K = K'}{\Gamma \vdash K' = K}(2.2)$. $\mathbf{l}^\circ_=(d) \equiv \mathbf{r}^\circ_=(d_0)$ et $\mathbf{r}^\circ_=(d) \equiv \mathbf{l}^\circ_=(d_0)$.

• Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} K = K' \qquad \Gamma \overset{d_2}{\vdash} K' = K''}{\Gamma \vdash K = K''}(2.3)$. $\mathbf{l}^\circ_=(d) \equiv \mathbf{l}^\circ_=(d_1)$ et $\mathbf{r}^\circ_=(d) \equiv \mathbf{r}^\circ_=(d_2)$.

• Cas $d \equiv \dfrac{\Gamma \overset{d_0}{\vdash} A = B : \textbf{Type}}{\Gamma \vdash El(A) = El(B)\textbf{kind}}(4.3)$. Par hypothèse d'induction $l^\circ_=(d_0)$ et $r^\circ_=(d_0)$ sont définis.

$$\mathbf{l}^\circ_=(d) \equiv \frac{\Gamma \overset{l^\circ_=(d_0)}{\vdash} A : \textbf{Type}}{\Gamma \vdash El(A)\textbf{kind}}(4.2) \quad \text{et} \quad \mathbf{r}^\circ_=(d) \equiv \frac{\Gamma \overset{r^\circ_=(d_0)}{\vdash} B : \textbf{Type}}{\Gamma \vdash El(B)\textbf{kind}}(4.2)$$

• Cas $d \equiv \dfrac{\Gamma, x : K_1 \overset{d_1}{\vdash} K'_1 = K'_2 \qquad \Gamma \overset{d_2}{\vdash} K_1 = K_2}{\Gamma \vdash (x : K_1)K'_1 = (x : K_2)K'_2}(5.2)$.

$$\mathbf{l}^\circ_=(d) \equiv \frac{\Gamma, x : K \overset{\mathbf{l}^\circ_=(d_1)}{\vdash} K'_1\textbf{kind}}{\Gamma \vdash (x : K_1)K'_1\textbf{kind}}(5.1) \quad \text{et}$$

$$\mathbf{r}^\circ_=(d) \equiv \mathbf{E}^\circ\Big(\dfrac{\Gamma, x : K_1 \overset{\mathbf{r}^\circ_=(d_1)}{\vdash} K_2'\mathbf{kind} \qquad \Gamma \overset{d_2}{\vdash} K_1 = K_2 \qquad \Gamma \overset{d_2}{\vdash} K_2\mathbf{kind}}{\dfrac{\Gamma, x : K_2 \vdash K_2'\mathbf{kind}}{\Gamma \vdash (x : K_2)K_2'\mathbf{kind}}\ 5.1}\ 3.3'\Big)$$

Pas d'induction pour le cas 2). Les dérivations suivantes se terminent par les règles : 2.5, 2.6, 3.2, 5.4, 5.6, 5.7, 5.8, CA.2, CD.
Les cas 2.5 et 2.6 sont similaires aux cas 2.2 et 2.3 avec $l^\circ_=$ et $r^\circ_=$ à la place de $\mathbf{l}^\circ_=$ et $\mathbf{r}^\circ_=$.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} k = k' : K \qquad \Gamma \overset{d_2}{\vdash} K = K'}{\Gamma \vdash k = k' : K'}\ (3.2)$

  $l^\circ_=(d) \equiv \dfrac{\Gamma \overset{l^\circ_=(d_1)}{\vdash} k : K \qquad \Gamma \overset{d_2}{\vdash} K = K'}{\Gamma \vdash k : K'}\ (3.1)$ et

  $r^\circ_=(d) \equiv \dfrac{\Gamma \overset{r^\circ_=(d_1)}{\vdash} k' : K \qquad \Gamma \overset{d_2}{\vdash} K = K'}{\Gamma \vdash k' : K'}\ (3.1)$

- Cas $d \equiv \dfrac{\Gamma, x : Q_1 \overset{d_1}{\vdash} q_1 = q_2 : K \qquad \Gamma \overset{d_2}{\vdash} Q_1 = Q_2}{\Gamma \vdash [x : Q_1]q_1 = [x : Q_2]q_2 : (x : Q_1)Q}\ (5.4)$

  avec $k_1 \equiv [x : Q_1]q_1, \quad k_2 \equiv [x : Q_2]q_2$ et $K_i \equiv (x : Q_1)Q$.

  $l^\circ_= \equiv \dfrac{\Gamma, x : Q_1 \overset{l^\circ_=(d_1)}{\vdash} q_1 : Q}{\Gamma \vdash [x : Q_1]q_1 : (x : Q_1)Q}\ (5.3)$

  Par hypothèse d'induction nous savons que $\mathbf{kd}^\circ(d_1)$ est définie. Nous avons :

$$r^\circ_=(d) \equiv \dfrac{\Gamma \overset{h}{\vdash} [x : Q_2]q_2 : (x : Q_2)Q \qquad \dfrac{\dfrac{\dfrac{\Gamma, x : Q_1 \overset{\mathbf{kd}^\circ(d_1)}{\vdash} Q\mathbf{kind}}{\Gamma, x : Q_1 \vdash Q = Q}\ 2.1 \qquad \Gamma \overset{d_2}{\vdash} Q_1 = Q_2}{\Gamma \vdash (x : Q_1)Q = (x : Q_2)Q}\ 5.2}{\Gamma \vdash (x : Q_2)Q = (x : Q_1)Q}\ 2.2}{\Gamma \vdash [x : Q_2]q_2 : (x : Q_2)Q}\ 3.1$$

  avec $h \equiv \mathbf{E}^\circ\Big(\dfrac{\Gamma, x : Q_1 \overset{r^\circ_=(d_1)}{\vdash} q_2 : Q \qquad \Gamma \overset{d_2}{\vdash} Q_1 = Q_2 \qquad \Gamma \overset{d_2}{\vdash} Q_2\mathbf{kind}}{\dfrac{\Gamma, x : Q_2 \vdash q_2 : Q}{\Gamma \vdash [x : Q_2]q_2 : (x : Q_2)Q}\ 5.3}\ 3.3'\Big)$

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f_1 = f_2 : (x:Q)Q' \qquad \Gamma \overset{d_2}{\vdash} q_1 = q_2 : K}{\Gamma \vdash f_1(q_1) = f_2(q_2) : [q_1/x]Q'}\,(5.6)\,.$

  avec $k_i \equiv f_i(q_i)$ et $K \equiv [q_1/x]Q'$.

  $$l^\circ_=(d) \equiv \dfrac{\Gamma \overset{l^\circ_=(d_1)}{\vdash} f_1 : (x:Q)Q' \qquad \Gamma \overset{l^\circ_=(d_2)}{\vdash} q_1 = q_2 : K}{\Gamma \vdash f_1(q_1) : [q_1/x]Q'}\,(5.5)$$

  Nous pouvons définir les dérivations $h \equiv \mathbf{kd}^\circ(d_1)$ par hypothèse d'induction et $h' \equiv p\_^\circ(h)$ par le lemme 4.13.

  $l^\circ_=(d) \equiv$

$$\dfrac{\dfrac{\Gamma \overset{r^\circ_=(d_1)}{\vdash} f_2 : (x:Q)Q' \qquad \Gamma \overset{r^\circ_=(d_2)}{\vdash} q_2 : Q}{\Gamma \vdash f_2(q_2) : [q_2/x]Q'}\,5.5 \qquad \dfrac{\mathbf{E}^\circ\Big(\dfrac{\Gamma, x:Q \overset{h'}{\vdash} Q'\,\mathbf{kind} \qquad \Gamma \overset{d_2}{\vdash} q_1 = q_2 : Q \qquad \Gamma \overset{l^\circ_=(d_2)}{\vdash} q_1 : Q \quad \Gamma \overset{r^\circ_=(d_2)}{\vdash} q_2 : Q}{\Gamma \vdash [q_1/x]Q' = [q_2/x]Q'}\,6.6'\Big)}{\Gamma \vdash [q_2/x]Q' = [q_1/x]Q'}\,2.2}{\Gamma \vdash f_2(q_2) : [q_1/x]Q'}\,3.1$$

- Cas $d \equiv \dfrac{\Gamma, x:Q \overset{d_1}{\vdash} q' : Q' \qquad \Gamma \overset{d_2}{\vdash} q : Q}{\Gamma \vdash ([x:Q]q')q = [q/x]q' : [q/x]Q'}\,(5.7)$

  avec $k_1 \equiv ([x:Q]q')q$, $k_2 \equiv [q/x]q'$ et $K \equiv [q/x]Q'$.

  $$l^\circ_=(d) \equiv \dfrac{\dfrac{\Gamma, x:Q \overset{d_1}{\vdash} q' : Q'}{\Gamma \vdash [x:Q]q' : (x:Q)Q'}\,(5.3) \qquad \Gamma \overset{d_2}{\vdash} q : Q}{\Gamma \vdash ([x:Q]q')q : [q/x]Q'}\,(5.5)$$

  et $r^\circ_=(d) \equiv \mathbf{E}^\circ\Big(\dfrac{\Gamma, x:Q \overset{d_1}{\vdash} q' : Q' \qquad \Gamma \overset{d_2}{\vdash} q : Q}{\Gamma, [k/x]\Gamma' \vdash [q/x]q' : [q/x]Q'}\,(6.3)\Big)$

- Cas $d \equiv \dfrac{\Gamma \overset{d_0}{\vdash} f : (x:Q)Q' \quad x \notin FV(\Gamma)}{\Gamma \vdash [x:Q]f(x) = f : (x:Q)Q'}\,(5.8)$

  avec $k_1 \equiv [x:Q]f(x)$, $k_2 \equiv f$ et $K \equiv (x:Q)Q'$.

$$l^\circ_=(d) \equiv \frac{\mathbf{E}^\circ\left(\dfrac{\Gamma \overset{d'_0}{\vdash} f : (y:Q)[y/x]Q' \qquad \Gamma, x:Q \overset{h'}{\vdash} \textbf{valid}}{\Gamma, x:Q \vdash f : (y:Q)[y/x]Q'}\,(\text{wkn})\right) \qquad \dfrac{\Gamma, x:Q \overset{h''}{\vdash} \textbf{valid}}{\Gamma, x:Q \vdash x : Q}\,(1.3)}{\dfrac{\Gamma, x:Q \vdash f(x) : Q'}{\Gamma \vdash [x:Q]f(x) : (x:Q)Q'}\,(5.3)}\,(5.5)$$

Et $r^\circ_=(d) \equiv d_0$

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f_1 = f_2 : (x:Q)Q' \qquad \Gamma \overset{d_2}{\vdash} q_1 = q_2 : Q_0 \qquad \Gamma \overset{d_3}{\vdash} Q_0 <_c Q}{\Gamma \vdash f_1(q_1) = f_2(q_2) : [c(q_1)/x]Q'}\,(\text{CA.2})$

  avec $k_1 \equiv f_1(q_1)$, $k_2 \equiv f_2(q_2)$ et $K \equiv [c(q_1)/x]Q'$.

  $$l^\circ_=(d) \equiv \frac{\Gamma \overset{l^\circ_=(d_1)}{\vdash} f_1 : (x:Q)Q' \qquad \Gamma \overset{l^\circ_=(d_2)}{\vdash} q_1 : Q_0 \qquad \Gamma \overset{d_3}{\vdash} Q_0 <_c Q}{\Gamma \vdash f_1(q_1) : [c(q_1)/x]Q'}\,(\text{CA.1})$$

  Par hypothèse d'induction nous savons que $h \equiv \Gamma \overset{\mathbf{kd}^\circ(d_1)}{\vdash} (x:Q)Q'$ est définie. De plus grâce au lemme 4.13, on a $\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (x:Q_0)Q$. Considérons la dérivation suivante :

$$h' \equiv \frac{\Gamma, x:Q \overset{p\_^\circ(h)}{\vdash} Q'\textbf{kind} \qquad \dfrac{\dfrac{\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (Q_0)Q}{\Gamma \vdash c = c : (Q_0)Q}\,(2.4) \qquad \Gamma \overset{d_2}{\vdash} q_1 = q_2 : Q_0}{\Gamma, x:Q \vdash c(q_1) = c(q_2) : Q}\,(5.6) \qquad \dfrac{\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (Q_0)Q \qquad \Gamma \overset{r^\circ_=(d_2)}{\vdash} q_2 : Q_0}{\Gamma, x:Q \vdash c(q_2) : Q}\,(5.5)}{\Gamma \vdash [c(q_1)/x]Q' = [c(q_2)/x]Q'}\,(6.6\text{'})$$

  Finalement on prend $r^\circ_=(d) \equiv$

$$r^\circ_=(d) \equiv \frac{\dfrac{\Gamma \overset{r^\circ_=(d_1)}{\vdash} f_2 : (x:Q)Q' \qquad \Gamma \overset{r^\circ_=(d_2)}{\vdash} q_2 : Q_0 \qquad \Gamma \overset{d_3}{\vdash} Q_0 <_c Q}{\Gamma \vdash f_1(q_1) : [c(q_1)/x]Q'}\,(\text{CA.1}) \qquad \dfrac{\Gamma \overset{\mathbf{E}^\circ(h')}{\vdash} [c(q_1)/x]Q' = [c(q_2)/x]Q'}{\Gamma \vdash [c(q_2)/x]Q' = [c(q_1)/x]Q'}\,(2.2)}{\Gamma \vdash f_2(q_2) : [c(q_1)/x]Q'}$$

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f = (x:Q)Q' \qquad \Gamma \overset{d_2}{\vdash} q : Q_0 \qquad \Gamma \overset{d_3}{\vdash} Q_0 <_c Q}{\Gamma \vdash f(q) = f(c(q)) : [c(q)/x]Q'}\,(\text{CD})$

  avec $k_1 \equiv f(q)$, $\quad k_2 \equiv f(c(q)) \quad$ et $\quad K \equiv [c(q)/x]Q'$

$$l^{\circ}_{=}(d) \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f = (x : Q)Q' \qquad \Gamma \overset{d_2}{\vdash} q : Q_0 \qquad \Gamma \overset{d_3}{\vdash} Q_0 <_c Q}{\Gamma \vdash f(q) : [c(q)/x]Q'} \text{ (CA.1)}$$

$$r^{\circ}_{=} \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f = (x : Q)Q' \qquad \dfrac{\Gamma \overset{\mathbf{co}^{\circ}(d_3)}{\vdash} q : (Q_0)Q \qquad \Gamma \overset{d_2}{\vdash} q : Q_0}{\Gamma \vdash c(q) : Q} \text{ (5.5)}}{\Gamma \vdash f(c(q)) : [c(q)/x]Q'} \text{ (5.5)}$$

Pas d'induction pour le cas 3). Voyons le cas des règles : 1.3, 2.4, 2.5, 2.6, 5.8, 3.1, 3.2, 5.3, 5.4, 5.5, 5.6 5.7 ST1 CA.1 CA.2 CD.

- Les règles (2.4, 2.5, 2.6, 5.8) se présentent de la forme : $d \equiv \dfrac{\ldots \Gamma \overset{d_0}{\vdash} \Sigma : K \ldots}{\Gamma \vdash \Sigma' : K}$
  En appliquant l'hypothèse d'induction nous obtenons $\mathbf{kd}^{\circ}(d) \equiv \mathbf{kd}^{\circ}(d_0)$.

- Cas $d \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K} \text{ (1.3)}$
  Le lemme 4.13 nous permet d'obtenir la dérivation :
  $$\mathbf{kd}^{\circ}(d) \equiv \mathbf{E}^{\circ}\left(\dfrac{\Gamma \overset{K^{\circ}_{\Gamma_1}(d_0)}{\vdash} K\textbf{kind} \qquad \Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash K\textbf{kind}} \text{ (wkn)}\right)$$

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} k : K \qquad \Gamma \overset{d_2}{\vdash} K = K'}{\Gamma \vdash k : K'} \text{ (3.1)}$. Par hypothèse d'induction nous avons la dérivation $\Gamma \overset{\mathbf{r}^{\circ}_{=}(d_2)}{\vdash} K'\textbf{kind}$. On prend $\mathbf{kd}^{\circ}(d) \equiv \mathbf{r}^{\circ}_{=}(d_2)$.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} k = k' : K \qquad \Gamma \overset{d_2}{\vdash} K = K'}{\Gamma \vdash k = k' : K'} \text{ (3.2)}$ Par hypothèse d'induction nous avons la dérivation $\Gamma \overset{\mathbf{r}^{\circ}_{=}(d_2)}{\vdash} K'\textbf{kind}$. On prend $\mathbf{kd}^{\circ}(d) \equiv \mathbf{r}^{\circ}_{=}(d_2)$.

- Cas $d \equiv \dfrac{\Gamma, x : K \overset{d_0}{\vdash} k : K'}{\Gamma \vdash [x : K]k : (x : K)K'}$ (5.3) · $\mathbf{kd}^\circ(d) \equiv \dfrac{\Gamma, x : K \overset{\mathbf{kd}^\circ(d_0)}{\vdash} K'\mathbf{kind}}{\Gamma \vdash (x : K)K'\mathbf{kind}}$ (5.1)

- Cas $d \equiv \dfrac{\Gamma, x : K_1 \overset{d_1}{\vdash} k_1 = k_2 : K \qquad \Gamma \overset{d_2}{\vdash} K_1 = K_2}{\Gamma \vdash [x : K_1]k_1 = [x : K_2]k_2 : (x : K_1)K}$ (5.4) on a

  $\mathbf{kd}^\circ(d) \equiv \dfrac{\Gamma, x : K_1 \overset{\mathbf{kd}^\circ(d_1)}{\vdash} K\mathbf{kind}}{\Gamma \vdash (x : K_1)K\mathbf{kind}}$ (5.1)

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f : (x : K)K' \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma \vdash f(k) : [k/x]K'}$ (5.5)

  Par hypothèse d'induction nous avons $\Gamma \overset{h}{\vdash} (x : K)K'\mathbf{kind}$, avec $h \equiv \mathbf{kd}^\circ(d_1)$, et le lemme 4.13 nous permet de définir :

  $\mathbf{kd}^\circ(d) \equiv \mathbf{E}^\circ\Big(\dfrac{\Gamma, x : K \overset{p\_^\circ(h)}{\vdash} K'\mathbf{kind} \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma \vdash [k/x]K'\mathbf{kind}}$ (6.2)$\Big)$

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f = f' : (x : K)K' \qquad \Gamma \overset{d_2}{\vdash} k_1 = k_2 : K}{\Gamma \vdash f(k_1) = f'(k_2) : [k_1/x]K'}$ (5.6)

  Par hypothèse d'induction nous avons $\Gamma \overset{h}{\vdash} (x : K)K'\mathbf{kind}$, avec $h \equiv \mathbf{kd}^\circ(d_1)$, et $\Gamma \overset{l^\circ_=(d_2)}{\vdash} k_1 : K$. Le lemme 4.13 nous permet de définir :

  $\mathbf{kd}^\circ(d) \equiv \mathbf{E}^\circ\Big(\dfrac{\Gamma, x : K \overset{p\_^\circ(h)}{\vdash} K'\mathbf{kind} \qquad \Gamma \overset{l^\circ_=(d_2)}{\vdash} k_1 : K}{\Gamma \vdash [k_1/x]K'\mathbf{kind}}$ (6.3)$\Big)$

- Cas $d \equiv \dfrac{\Gamma, x : K \overset{d_1}{\vdash} k' : K' \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma \vdash ([x : K]k')k = [k/x]k' : [k/x]K'}$ (5.7)

$$\mathbf{kd}^\circ(d) \equiv \mathbf{E}^\circ(\frac{\Gamma, x : K \overset{\mathbf{kd}^\circ(d_1)}{\vdash} K'\mathbf{kind} \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma \vdash [k/x]K'\mathbf{kind}} (6.3))$$

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f : (x : K)K' \qquad \Gamma \overset{d_2}{\vdash} k : K_0 \qquad \Gamma \overset{d_3}{\vdash} K_0 <_c K}{\Gamma \vdash f(k) : [c(k)/x]K'}$ (CA.1)

  Par hypothèse d'induction nous avons $\Gamma \overset{h}{\vdash} (x : K)K'\mathbf{kind}$, avec $h \equiv \mathbf{kd}^\circ(d_1)$. Le lemme 4.13 nous permet de définir :

$$\mathbf{kd}^\circ(d) \equiv \quad \mathbf{E}^\circ(\frac{\Gamma, x : K \overset{p\_^\circ(h)}{\vdash} K'\mathbf{kind} \qquad \dfrac{\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (K_0)K \qquad \Gamma \overset{d_2}{\vdash} k : K_0}{\Gamma \vdash c(k) : K} (5.5)}{\Gamma \vdash [c(k)/x] : K'} (6.2))$$

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f = f' : (x : K)K' \qquad \Gamma \overset{d_2}{\vdash} k_1 = k_2 : K_0 \qquad \Gamma \overset{d_3}{\vdash} K_0 <_c K}{\Gamma \vdash f(k_1) = f'(k_2) : [c(k_1)/x]K'}$ (CA.2)

  Par hypothèse d'induction nous avons $\Gamma \overset{h}{\vdash} (x : K)K'\mathbf{kind}$, avec $h \equiv \mathbf{kd}^\circ(d_1)$, et $\Gamma \overset{l^\circ_=(d_2)}{\vdash} k_1 : K_0$. Le lemme 4.13 nous permet de définir :

$$\mathbf{kd}^\circ(d) \equiv \quad \mathbf{E}^\circ(\frac{\Gamma, x : K \overset{p\_^\circ(h)}{\vdash} K'\mathbf{kind} \qquad \dfrac{\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (K_0)K \qquad \Gamma \overset{l^\circ_=(d_2)}{\vdash} k_1 : K_0}{\Gamma \vdash c(k_1) : K} (5.5)}{\Gamma \vdash [c(k_1)/x] : K'} (6.2))$$

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f : (x : K)K' \qquad \Gamma \overset{d_2}{\vdash} k_0 : K_0 \qquad \Gamma \overset{d_3}{\vdash} K_0 <_c K}{\Gamma \vdash f(k_0) : [c(k_0)/x]K'}$ (CD)

  Par hypothèse d'induction nous avons $\Gamma \overset{h}{\vdash} (x : K)K'\mathbf{kind}$, avec $h \equiv \mathbf{kd}^\circ(d_1)$. Le lemme 4.13 nous permet de définir :

$$\mathbf{kd}^\circ(d) \equiv \quad \mathbf{E}^\circ(\frac{\Gamma, x : K \overset{p\_^\circ(h)}{\vdash} K'\mathbf{kind} \qquad \dfrac{\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (K_0)K \qquad \Gamma \overset{d_2}{\vdash} k_0 : K_0}{\Gamma \vdash c(k_0) : K}\,(5.5)}{\Gamma \vdash [c(k_0)/x] : K'}\,(6.2))$$

□

Lemme 4.21. *Il existe des algorithmes* ***less*** *et* ***more*** *qui transforment respectivement toute* $UTT^r[R]^-_w$*-dérivation* d *en une* $UTT^r[R]^-$*-dérivation, et toute* $UTT^r[R]^-$*-dérivation* d' *en une* $UTT^r[R]^-_w$*-dérivation, tels que* ***less***$(d) \sim d$ *et* ***more***$(d') \sim d'$ *respectivement. Si* d *est une* $UTT^r[R]^-$*-dérivation alors* ***less***$($***more***$(d)) \equiv d$*.*

Démonstration. L'algorithme **less** efface juste les prémisses supplémentaires, des règles de $UTT^r[R]^-_w$ qui ne sont pas des règles de $UTT^r[R]^-$, des dérivations. Tandis que l'algorithme **more** est défini par induction sur le nombre de règles de $UTT^r[R]^-$ qui ont une forme différente dans $UTT^r[R]^-_w$.
Cas de base. Quand il n'y a aucune des règles considérées ci dessus : **more**$(d) \equiv d$.
Pas d'induction.
exemple cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} K_1 <_c K_2 \qquad \Gamma \overset{d_2}{\vdash} K_2 = K_2'}{\Gamma \vdash K_1 <_c K_2'}$ (SK.5)
En appliquant l'hypothèse d'induction, on sait que **more**(d_1) et **more**(d_2) sont définies et sont dans $UTT^r[R]^-_w$. On peut donc rajouter la prémisse supplémentaire $\Gamma \overset{\mathbf{r}^\circ_<(\mathbf{more}(d_2))}{\vdash} K_2'\mathbf{kind}$ et obtenir

$$\mathbf{more}(d) \equiv \frac{\Gamma \overset{\mathbf{more}(d_1)}{\vdash} K_1 <_c K_2 \qquad \Gamma \overset{\mathbf{more}(d_2)}{\vdash} K_2 = K_2' \quad \Gamma \overset{\mathbf{r}^\circ_<(\mathbf{more}(d_2))}{\vdash} K_2'\mathbf{kind}}{\Gamma \vdash K_1 <_c K_2'}\ \text{(SK.5')}$$

□

Lemme 4.22. *Il existe un algorithme* ***E***** *qui transforme toute* $UTT^r[R]$*-dérivation* d *en une* $UTT^r[R]^-_w$*-dérivation du même jugement.*

Démonstration. Preuve par induction sur le nombre de règles de $UTT^r[R]$ qui ne sont pas des règles de $UTT^r[R]^-_w$.
Cas de base. Il n'existe aucune de ces règles dans d. Dans ce cas $\mathbf{E}^*(d) \equiv d$.

Pas d'induction.

- Cas où la dernière règle de d est une des règles dont il existe une version avec une prémisse supplémentaire.

  Exemple cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} K_1 <_c K_2 \qquad \Gamma \overset{d_2}{\vdash} K_2 = K_2'}{\Gamma \vdash K_1 <_c K_2'}$ (SK.5) .

  En appliquant l'hypothèse d'induction, on sait que $\mathbf{E}^*(d_1)$ et $\mathbf{E}^*(d_2)$ sont définies et sont dans $UTT^r[R]^-_w$. On peut donc rajouter la prémisse supplémentaire $\Gamma \overset{\mathbf{r}^\circ_<(\mathbf{E}^*(d_2))}{\vdash} K_2'\mathbf{kind}$ et obtenir :

$$\mathbf{E}^*(d) \equiv \frac{\Gamma \overset{\mathbf{E}^*(d_1)}{\vdash} K_1 <_c K_2 \qquad \Gamma \overset{\mathbf{E}^*(d_2)}{\vdash} K_2 = K_2' \quad \Gamma \overset{\mathbf{r}^\circ_<(\mathbf{E}^*(d_2))}{\vdash} K_2'\mathbf{kind}}{\Gamma \vdash K_1 <_c K_2'} \text{(SK.5')}$$

- Cas où la dernière règle de d est une des règles à éliminer.

  Considérons le cas $d \equiv \dfrac{\Gamma_1, \Gamma_2 \overset{d_1}{\vdash} J \qquad \Gamma_1, \Gamma_3 \overset{d_2}{\vdash} \mathbf{valid}}{\Gamma_1, \Gamma_3\Gamma_2 \vdash J}$ (wkn)

  En appliquant l'hypothèse d'induction, on sait que $\mathbf{E}^*(d_1)$ et $\mathbf{E}^*(d_2)$ sont définies et sont dans $UTT^r[R]^-_w$. Les conditions sont réunies pour que l'on puisse utiliser l'algorithme $\mathbf{E}^\circ_{wkn}$ afin d'éliminer wkn.

  On obtient $\mathbf{E}^*(d) \equiv \mathbf{E}^\circ_{wkn}\left(\dfrac{\Gamma_1, \Gamma_2 \overset{\mathbf{E}^*(d_1)}{\vdash} J \qquad \Gamma_1, \Gamma_3 \overset{\mathbf{E}^*(d_2)}{\vdash} \mathbf{valid}}{\Gamma_1, \Gamma_3\Gamma_2 \vdash J} \text{(wkn)}\right)$.

□

Lemme 4.23. *Il existe un algorithme* **E** *qui transforme toute $UTT^r[R]$-dérivation d en une $UTT^r[R]^-$-dérivation du même jugement.*

Démonstration. Quelque soit la dérivation d, $\mathbf{E}(d) \equiv \mathbf{less}(\mathbf{E}^*(d))$.

□

Theorème 4.24. *Il existe des algorithmes V_{Γ_1}, K_{Γ_1}, **l**$_=$, **r**$_=$, $l_=$, $r_=$, **kd**, **l**$_<$, **r**$_<$ et **co** qui transforment toute $UTT^r[R]$-dérivation d de*

*1) $\Gamma_1, \Gamma_2 \vdash J$ en une $UTT^r[R]^-$-dérivation $V_{\Gamma_1}(d)$ de $\Gamma_1 \vdash$ **valid** ;*

*2) $\Gamma_1, x : K, \Gamma_2 \vdash J$ en une $UTT^r[R]^-$-dérivation $K_{\Gamma_1}(d)$ de $\Gamma_1 \vdash K$**Kind** ;*

*3) $\Gamma \vdash K_1 = K_2$ en deux $UTT^r[R]^-$-dérivations **l**$_=(d)$ de $\Gamma \vdash K_1$**kind** et **r**$_=(d)$ de $\Gamma \vdash K_2$**kind** ;*

4) $\Gamma \vdash k_1 = k_2 : K$ en deux $UTT^r[R]^-$-dérivations $l_=(d)$ de $\Gamma \vdash k_1 : K$ et $r_=(d)$ de $\Gamma \vdash k_2 : K$;

*5) $\Gamma \vdash \Sigma : K$ en une $UTT^r[R]^-$-dérivation **kd**(d) de $\Gamma \vdash K$**kind** (Σ dénote un terme ou une égalité entre termes) ;*

*6) $\Gamma \vdash A <_c B :$ **Type** ou $\Gamma \vdash K <_c K'$ en des $UTT^r[R]^-$-dérivations :*

- ***l**$_<(d)$ de $\Gamma \vdash A :$ **Type** ou $\Gamma \vdash K$**kind**,*
- ***r**$_<(d)$ de $\Gamma \vdash B :$ **Type** ou $\Gamma \vdash K'$**kind**,*
- ***co**(d) de $\Gamma \vdash c : (El(A))El(B)$ ou $\Gamma \vdash c : (K)K'$ respectivement.*

Démonstration. Nous définissons les algorithmes en question de la manière suivante :
$V_{\Gamma_1}(d) \equiv V^\circ_{\Gamma_1}(\mathbf{E}(d))$, $K_{\Gamma_1}(d) \equiv K^\circ_{\Gamma_1}(\mathbf{E}(d))$, $\mathbf{l}_=(d) \equiv \mathbf{less}(\mathbf{l}^\circ_=(\mathbf{E}^*(d)))$, $\mathbf{r}_=(d) \equiv \mathbf{less}(\mathbf{r}^\circ_=(\mathbf{E}^*(d)))$, $l_=(d) \equiv \mathbf{less}(l^\circ_=(\mathbf{E}^*(d)))$, $r_=(d) \equiv \mathbf{less}(r^\circ_=(\mathbf{E}^*(d)))$, $\mathbf{kd}(d) \equiv \mathbf{less}(\mathbf{kd}^\circ(\mathbf{E}^*(d)))$, $\mathbf{l}_<(d) \equiv \mathbf{less}(\mathbf{l}^\circ_<(\mathbf{E}^*(d)))$, $\mathbf{r}_<(d) \equiv \mathbf{less}(\mathbf{r}^\circ_<(\mathbf{E}^*(d)))$, $\mathbf{co}(d) \equiv \mathbf{less}(\mathbf{co}^\circ(\mathbf{E}^*(d)))$.

□

Lemme 4.25.

*a) Il existe des algorithmes **P**_ et **P**'_ qui transforment toute $UTT^r[R]$-dérivation d de $\Gamma \vdash (x : K)Q$**kind** en des $UTT^r[R]^-$-dérivations de $\Gamma \vdash K$**kind** et $\Gamma, x : K \vdash Q$**kind** respectivement.*

b) *Il existe des algorithmes* $\mathbf{P}_=$, $\mathbf{P}'_=$ *et* $\mathbf{P}''_=$ *qui transforment toute* $UTT^r[R]$*-dérivation* d *de* $\Gamma \vdash (x:K)Q = (x:K')Q'$ *en des* $UTT^r[R]$*-dérivations de* $\Gamma \vdash K = K'$, $\quad \Gamma, x:K \vdash Q = Q' \quad$ *et* $\quad \Gamma, x:K' \vdash Q = Q'$ *respectivement.*

Démonstration.

a) On a $\mathbf{P}_\_(d) \equiv K^\circ_{\Gamma_1}(p\_{}^\circ(\mathbf{E}(d)))$ et $\mathbf{P}'_\_(d) \equiv p\_{}^\circ(\mathbf{E}(d))$.

b) Prenons $d' \equiv \mathbf{E}(d)$. On procède par induction sur la taille de d'.
Cas de base. C'est le cas de la règle 2.1 (voir développement plus bas).

Pas d'induction. les règles à considérer sont 2.1, 2.2, 2.3, 5.2.

- Considérons le cas $d' \equiv \dfrac{\Gamma, x:K \overset{d'_1}{\vdash} Q = Q' \qquad \Gamma \overset{d'_2}{\vdash} K = K'}{\Gamma \vdash (x:K)Q = (x:K')Q'}$ (5.2)

  Nous avons $\mathbf{P}_=(d') \equiv d'_2$, $\quad \mathbf{P}'_=(d') \equiv d'_1$ et $\mathbf{P}''_=(d') \equiv \dfrac{\Gamma, x:K \overset{d'_1}{\vdash} Q = Q' \qquad \Gamma \overset{d'_2}{\vdash} K = K'}{\Gamma, x:K' \vdash Q = Q'}$ (3.3)

- Considérons le cas

$$d' \equiv \dfrac{\Gamma \overset{d'_1}{\vdash} (x:K)Q = (x:K")Q" \qquad \Gamma \overset{d'_2}{\vdash} (x:K")Q" = (x:K')Q'}{\Gamma \vdash (x:K)Q = (x:K')Q'} \text{ (2.3)}$$

  – Nous pouvons appliquer l'hypothèse d'induction aux dérivations d'_1 et d'_2.
  Ensuite nous obtenons $\mathbf{P}_=(d') \equiv \dfrac{\Gamma \overset{\mathbf{P}_=(d'_1)}{\vdash} K = K" \qquad \Gamma \overset{\mathbf{P}_=(d'_2)}{\vdash} K" = K'}{\Gamma \vdash K = K'}$ (2.3)

  – Ci-dessous nous pouvons appliquer l'hypothèse d'induction aux prémisses de d'. Nous obtenons : $\mathbf{P}'_=(d') \equiv$

$$\dfrac{\Gamma, x:K \overset{\mathbf{P}'_=(d'_1)}{\vdash} Q = Q" \qquad \dfrac{\Gamma, x:K" \overset{\mathbf{P}'_=(d'_2)}{\vdash} Q" = Q' \qquad \dfrac{\Gamma \overset{\mathbf{P}_=(d'_1)}{\vdash} K = K"}{\Gamma \vdash K" = K} \text{ (2.2)}}{\Gamma, x:K \vdash Q" = Q'} \text{ (3.3)}}{\Gamma, x:K \vdash Q = Q'} \text{ (2.3)}$$

– Ci-dessous nous pouvons appliquer l'hypothèse d'induction aux prémisses de d'. Nous obtenons :

$$\mathbf{P}''_{=}(d') \equiv \cfrac{\cfrac{\Gamma, x : K" \overset{\mathbf{P}''_{=}(d'_1)}{\vdash} Q = Q" \qquad \Gamma, x : K" \overset{\mathbf{P}'_{=}(d'_2)}{\vdash} Q" = Q'}{\Gamma, x : K" \vdash Q = Q'}\,(2.3) \qquad \Gamma \overset{\mathbf{P}_{=}(d'_2)}{\vdash} K" = K'}{\Gamma, x : K' \vdash Q = Q'}\,(3.3)$$

- Considérons le cas $d' \equiv \cfrac{\Gamma \overset{d'_0}{\vdash} (x : K')Q' = (x : K)Q}{\Gamma \vdash (x : K)Q = (x : K')Q'}\,(2.2)$

  On obtient les dérivations suivantes :

$$\mathbf{P}_{=}(d') \equiv \cfrac{\Gamma \overset{\mathbf{P}_{=}(d'_0)}{\vdash} K' = K}{\Gamma \vdash K = K'}\,(2.2)$$

$$\mathbf{P}'_{=}(d') \equiv \cfrac{\cfrac{\Gamma, x : K' \overset{\mathbf{P}'_{=}(d'_0)}{\vdash} Q' = Q}{\Gamma, x : K' \vdash Q = Q'}\,(2.2) \qquad \Gamma \overset{\mathbf{P}_{=}(d'_0)}{\vdash} K' = K}{\Gamma, x : K \vdash Q = Q'}\,(3.3)$$

$$\mathbf{P}''_{=}(d') \equiv \cfrac{\Gamma, x : K' \overset{\mathbf{P}_{=}(d'_0)}{\vdash} Q' = Q}{\Gamma, x : K' \vdash Q = Q'}\,(2.2)$$

- Considérons le cas $d' \equiv \cfrac{\Gamma \overset{d'_0}{\vdash} (x : K)Q\,\mathbf{kind}}{\Gamma \vdash (x : K)Q = (x : K)Q}\,(2.1)$

  On obtient les dérivations suivantes :

$$\mathbf{P}_{=}(d') \equiv \cfrac{\Gamma \overset{K^{\circ}_{\Gamma_1}(p\_^{\circ}((d'_0)))}{\vdash} K\,\mathbf{kind}}{\Gamma \vdash K = K}\,(2.1) \quad \text{et}$$

$$\mathbf{P}'_{=}(d') \equiv \mathbf{P}''_{=}(d') \equiv \cfrac{\Gamma, x : K \overset{p\_^{\circ}((d'_0))}{\vdash} Q\,\mathbf{kind}}{\Gamma, x : K \vdash Q = Q}\,(2.1)$$

$\square$

Lemme 4.26. *Considérons une $UTT^r[R]$-dérivation d du jugement $\Gamma \vdash f(k)$: **Type**.*

1. *Il existe un algorithme **coer** qui établit si le terme $f(k)$ provient d'une application coercitive ou pas.*

2. *Si l'application n'est pas coercitive, il existe des algorithmes $\textbf{kl}_1$ et $\textbf{kl}_2$ qui en extraient des dérivations des jugements $\Gamma \vdash f : (K)$**Type** et $\Gamma \vdash k : K$ respectivement.*

3. *Si l'application est coercitive, il existe également un algorithme $\textbf{kl}_3$ tel que : $\textbf{kl}_1$ en extrait une dérivation du jugement $\Gamma \vdash f : (K)$**Type**, $\textbf{kl}_2$ en extrait une dérivation jugement $\Gamma \vdash k : K_0$ pour une sorte K_0 quelconque, et $\textbf{kl}_3$ en extrait une dérivation du jugement $\Gamma \vdash K_0 <_c K$.*

Démonstration.

1. Par observation de la dérivation d. Pour toute règle sauf 5.5 et CA.1, il existe une prémisse unique qui contient la "pré-image" de $f(k)$. En remontant la dérivation d, si l'on rencontre en premier l'application de la règle 5.5 avec les premisses $\Gamma \overset{d_1}{\vdash} f : (K)\textbf{Type}$ et $\Gamma \overset{d_2}{\vdash} k : K$, alors le terme $f(k)$ provient d'une application non coercitive. Si l'on rencontre en premier l'application de la règle CA.1 avec les prémisses $\Gamma \overset{d_1}{\vdash} f : (K)\textbf{Type}$, $\Gamma \overset{d_2}{\vdash} k : K_0$ et $\Gamma \overset{d_3}{\vdash} K_0 <_c K$, alors le terme $f(k)$ provient d'une application coercitive.

2. Par induction sur les dérivations. Cas où l'application est non coercitive.
   Cas de base. Lorsque d se termine par la règle 5.5.

   $$d \equiv \frac{\Gamma \overset{d_1}{\vdash} f : (K)\textbf{Type} \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma \vdash f(k) : \textbf{Type}} \; (5.5)$$

   on a $\textbf{kl}_1(d) \equiv d_1$, $\textbf{kl}_2(d) \equiv d_2$ et $\textbf{kl}_3(d)$ n'existe pas.

   Pas d'induction. Les règles à considérer sont 3.1 et 5.5. Voyons le cas 3.1.

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} f(k) : \textbf{Type} \qquad \Gamma \overset{d_2}{\vdash} \textbf{Type} = \textbf{Type}}{\Gamma \vdash f(k) : \textbf{Type}} \ (3.1)$$

Remarquons que la seule sorte pouvant être égale à **Type** est **Type** elle-même. On peut conclure par hypothèse d'induction sur d_1.

3. Par induction sur les dérivations. Cas où l'application est coercitive.
   Cas de base. Lorsque d se termine par la règle CA.1.

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} f : (K)\textbf{Type} \qquad \Gamma \overset{d_2}{\vdash} k : K_0 \qquad \Gamma \overset{d_3}{\vdash} K_0 <_c K}{\Gamma \vdash f(k) : \textbf{Type}} \ (\text{CA.1})$$

   on a $\mathbf{kl}_1(d) \equiv d_1, \quad \mathbf{kl}_2(d) \equiv d_2 \quad$ et $\mathbf{kl}_3(d) \equiv d_3$.

   Pas d'induction. Identique au cas 2 en considérant les règles 3.1 et CA.1.

□

Lemme 4.27. *Il existe un algorithme K_{Θ_i} qui transforme toute $UTT^r[R]$-dérivation d de $\Gamma \vdash \kappa^X[\bar{\Theta}] : K$ en des dérivations de $\Gamma, X :$ **Type** $\vdash \Theta_i$**kind** ($1 \leq i \leq n$). Avec $\bar{\Theta}$ de la forme $< \Theta_1, \ldots, \Theta_n >$, où $\kappa^X[\bar{\Theta}]$ est une expression constante.*

Démonstration. Par induction sur les dérivations.

□

4.2.3.4 Elimination de la transitivité pour les sous-sortes

Afin de pouvoir propager les conditions de cohérence qui sont vérifiées pour le sous-typage aux sous-sortes, nous cherchons ici à éliminer les règles de transitivité pour les sous-sortes dans le calcul $UTT^r[R]^-_{ok}$.

Lemme 4.28. *Il existe un algorithme El^- qui transforme toute $UTT^r[R]^-$-dérivation d de :*

*1) $\Gamma \vdash El(A)$**kind** en une dérivation de $\Gamma \vdash A :$ **Type** ;*

2) $\Gamma \vdash El(A) = El(B)$ *en une dérivation de* $\Gamma \vdash A = B$: ***Type*** *;*

3) $\Gamma \vdash El(A) <_c El(B)$ *en une dérivation de* $\Gamma \vdash A <_c B$: ***Type****.*

Démonstration. Preuve par induction sur la taille de d. Les règles à considérer sont 2.1, 2.2, 2.3, 4.2, 4.3,SK.1, SK.5, SK.6, SK.7, SK.8.
Cas de base.

1) Pour le cas de base, la dernière règle de d est (4.2). Le résultat est immédiat, en prenant la dérivation de la prémisse de cette dernière règle pour $El^{-}(d)$.
2) Ici la dernière règle de d est forcément (4.3). Le résultat est immédiat en prenant la dérivation de la prémisse de cette dernière règle pour $El^{-}(d)$.
3) Ici la dernière règle de d est forcément (SK.1). Le résultat est immédiat en prenant la dérivation de la prémisse de cette dernière règle pour $El^{-}(d)$.

Pas d'induction.
Pour les cas 4.2, 4.3,SK.1, la prémisse contient exactement le jugement recherché, on prend la dérivation de la prémisse pour $El^{-}(d)$, comme illustré dans les cas de base. Pour les autres règles (2.1, 2.2, 2.3, 4.2, 4.3,SK.1, SK.5, SK.6, SK.7, SK.8), on applique l'algorithme El^{-} aux dérivations des prémisses puis on applique la règle correspondante pour les types, à savoir : $2.2 \to 2.4$, $2.3 \to 2.6$, $SK.5 \to ST.3$, $SK.6 \to ST.2$, $SK.7 \to ST.4$, $SK.8 \to ST.5$. Prenons comme exemple :
$d' \equiv \dfrac{\Gamma \overset{d_1}{\vdash} El(A) <_c El(B) \qquad \Gamma \overset{d_2}{\vdash} c = c' : (El(A))El(B)}{\Gamma \vdash El(A) <_{c'} El(B)}$ (SK.8) . On obtient la dérivation suivante :

$$El^{-}(d) \equiv \frac{\Gamma \overset{El^{-}(d_1)}{\vdash} A <_c B \qquad \Gamma \overset{d_2}{\vdash} c = c' : (El(A))El(B)}{\Gamma \vdash A <_{c'} B} \text{ (ST.5)}$$

□

Définition 4.29 (Pré-sortes)**.** *Nous appelons pré-sortes, les expressions définies inductivement de la manière suivante :*

- ***Type*** *est une pré-sorte ;*

- *Si K et K' sont des pré-sortes alors $(x : K)K'$ est une pré-sorte. Toutes les occurrences de x dans K' sont liées.*

Définition 4.30. *Nous définissons le rang d'une pré-sorte, noté $rank(K)$, où K est une pré-sorte.*

- *Si $K \equiv El(k)$ ou $K \equiv$ **Type** alors $rank(K) = 0$;*
- *Si $K \equiv (x : K_1)K_2$ alors $rank(K) = max(rank(K_1), rank(K_2)) + 1$.*

Notons que le rang est défini pour toutes les pré-sortes.

Lemme 4.31. *Toutes les sortes de $UTT^r[R]$ sont des pré-sortes.*

Démonstration. Preuve par induction sur les dérivations.

□

Lemme 4.32. *Si K est une pré-sorte , k une expression quelconque et x une variable alors $rank(K) = rank([k/x]K)$.*

Démonstration. Par induction sur la taille de K.
Cas de base.
Quand $K \equiv El(F(x))$, on sait que $[k/x]El(F(x)) = El(F(k))$ et par définition $rank(El(F(x))) = rank(El(F(k))) = 0$.
Quand $K \equiv$ **Type**, on sait que $[k/x]$**Type** $=$ **Type** d'où $rank([k/x]\textbf{Type}) = rank(\textbf{Type}) = 0$.

Pas d'induction.
Prenons $K \equiv (x : K_1)K_2$, on sait que $[k/x]((x : K_1)K_2) = (x : [k/x]K_1)[k/x]K_2$. Par définition $rank([k/x]((x : K_1)K_2)) = max(rank([k/x]K_1), rank([k/x]K_2)) + 1$, or par hypothèse d'induction $rank([k/x]K_1) = rank(K_1)$ et $rank([k/x]K_2) = rank(K_2)$. Par conséquent $rank([k/x]((x : K_1)K_2)) = max(rank(K_1), rank(K_2)) + 1 = rank((x : K_1)K_2)$.

□

Lemme 4.33. *Dans le système $UTT^r[R]^-_{ok}$, on a :*

a) si $\Gamma \vdash K = K'$ est dérivable alors $rank(K) = rank(K')$;

b) si $\Gamma \vdash K <_c K'$ est dérivable sans les règles SK.5, SK.6 et SK.7, alors $rank(K) = rank(K')$.

Démonstration. Soit $r = max(rank(K), rank(K'))$, grâce au lemme 4.10, on sait que si une des deux sortes K ou K' est une sorte-produit alors l'autre l'est également.

a) On procède par induction sur r.
Cas de base . Quand $r = 0$, $rank(K) = rank(K') = 0$.
Pas d'induction. Si on a le jugement $\Gamma \vdash (x : Q_1)Q_2 = (x : Q'_1)Q'_2$, alors par rapport au lemme 4.25, on sait que les jugements $\Gamma \vdash Q_1 = Q'_1$ et $\Gamma, x : Q_1 \vdash Q_2 = Q'_2$ sont dérivables. $max(rank(Q_1), rank(Q'_1))$ et $max(rank(Q_1), rank(Q'_1))$ sont plus petits que $max(rank((x : Q_1)Q_2), rank((x : Q'_1)Q'_2))$, on peut ainsi appliquer l'hypothèse d'induction et on a $rank(Q_1) = rank(Q'_1)$ et $rank(Q_2) = rank(Q'_2)$. Par conséquent $rank((x : Q_1)Q_2) = max(rank(Q_1), rank(Q_2))+1 = max(rank(Q'_1), rank(Q'_2))+1 = rank((x : Q'_1)Q'_2)$.

b) Soit d une dérivation du jugement $\Gamma \vdash K <_c K'$, prenons s sa taille. Nous procédons par induction sur $r\omega + s$.
Cas de base . Quand $r = 0$, $rank(K) = rank(K') = 0$.
Pas d'induction. Le jugement final de d est $\Gamma \vdash (x : Q_1)Q_2 <_c (x : Q'_1)Q'_2$, avec $K \equiv (x : Q_1)Q_2$ et $K' \equiv (x : Q'_1)Q'_2$. On présume que les règles SK.5, SK.6 et SK.7 n'apparaissent pas dans d. Les règles à considérer sont SK.2, SK.3, SK.4, SK.8.

- Considérons le cas où la dernière de d est SK.2 (cas similaire pour SK.3 et SK.4).

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} Q'_1 <_{c_1} Q_1 \qquad \Gamma, x' : Q'_1 \overset{d_2}{\vdash} [c_1(x')/x]Q_2 = Q'_2 \qquad \Gamma, x : Q_1 \overset{d_3}{\vdash} Q_2 \mathbf{kind}}{\Gamma \vdash (x : Q_1)Q_2 <_c (x' : Q'_1)Q'_2} \text{(SK.2)}$$

  Étant donné que $max(rank(Q_1), rank(Q'_1))$ est plus petit que $max(rank((x : Q_1)Q_2), rank((x : Q'_1)Q'_2))$, on peut ainsi appliquer l'hypothèse d'induction et on a $rank(Q_1) = rank(Q'_1)$. De plus grâce au a) du lemme et au lemme 4.28 on sait que $rank(Q_2) = rank(Q'_2)$. On obtient donc que

$rank((x:Q_1)Q_2) = max(rank(Q_1), rank(Q_2)) + 1 = max(rank(Q'_1), rank(Q'_2)) + 1 = rank((x:Q'_1)Q'_2)$.

- Considérons le cas

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} (x:Q_1)Q_2 <_c (x:Q'_1)Q'_2 \qquad \Gamma \overset{d_2}{\vdash} c = c' : ((x:Q_1)Q_2)(x:Q'_1)Q'_2}{\Gamma \vdash (x:Q_1)Q_2 <_{c'} (x:Q'_1)Q'_2} \text{(SK.8)}$$

Ici le paramètre r reste inchangé au niveau des prémisses, par contre le paramètre s diminue. On peut donc appliquer l'hypothèse d'induction au jugement final de d_1 et obtenir $rank((x:Q_1)Q_2) = rank((x:Q'_1)Q'_2)$.

□

Lemme 4.34. *Il existe un algorithme qui transforme toute $UTT^r[R]^-_{ok}$-dérivation d de la forme :*

$$\frac{\Gamma \overset{d_1}{\vdash} K <_c K' \qquad \Gamma \overset{d_2}{\vdash} K' = K''}{\Gamma \vdash K <_c K''} \textit{(SK.5)} \quad \textit{ou} \quad \frac{\Gamma \overset{d_1}{\vdash} K' <_c K'' \qquad \Gamma \overset{d_2}{\vdash} K' = K}{\Gamma \vdash K <_c K''} \textit{(SK.6)}$$

où d_1 et d_2 ne contiennent pas les règles de transitivité SK.5, SK.6 et SK.7 en une dérivation d' du même jugement qui ne contient pas les règles SK.5, SK.6 et SK.7.

Démonstration. Grâce au lemme 4.33, on sait que $rank(K) = rank(K') = rank(K'') = r$. Soit s la taille de la dérivation d_1, la preuve se fait par induction sur $r\omega + s$.

Cas de base. Quand r=0, $K \equiv El(A)$, $K' \equiv El(A')$, $K'' \equiv El(A'')$. Les démonstrations pour les cas SK.5 et SK.6 sont similaires.

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} El(A) <_c El(A') \qquad \Gamma \overset{d_2}{\vdash} El(A') = El(A'')}{\Gamma \vdash El(A) <_c El(A'')} \text{(SK.5)}$$

En utilisant le lemme 4.28, on obtient la dérivation :

$$\frac{\dfrac{\Gamma \overset{El^-(d_1)}{\vdash} A <_c A' : \mathbf{Type} \qquad \Gamma \overset{El^-(d_2)}{\vdash} A' = A'' : \mathbf{Type}}{\Gamma \vdash A <_c A'' : \mathbf{Type}} \text{(ST.3)}}{\Gamma \vdash El(A) <_c El(A'')} \text{(SK.1)}$$

Pas d'induction. La dérivation d_1 peut se terminer par les règles suivantes : SK.2, SK.3, SK.4, SK.8.

- Considérons ici le cas où d_1 se termine par la règle SK.8. Les démonstrations quand d se termine par SK.5 ou SK.6 sont similaires. On considère le cas de la règle SK.5.

  Cas $d \equiv$

$$\dfrac{\dfrac{\Gamma \overset{d_1}{\vdash} K <_{c'} K' \qquad \Gamma \overset{d_2}{\vdash} c' = c : (K)K'}{\Gamma \vdash K <_c K'}\,\text{(SK.8)} \qquad \Gamma \overset{d_2}{\vdash} K' <_c K''}{\Gamma \vdash K <_c K''}\,\text{(SK.5)}$$

  On obtient :

$$\dfrac{\dfrac{\Gamma \overset{d_1}{\vdash} K <_{c'} K' \qquad \Gamma \overset{d_2}{\vdash} K' = K''}{\Gamma \vdash K <_{c'} K''}\,\text{(SK.5)} \qquad \dfrac{\Gamma \overset{d_1}{\vdash} c' = c : (K)K' \qquad \Gamma \overset{d_2}{\vdash} (K)K' = (K)K''}{\Gamma \vdash c' = c : (K)K''}\,\text{(3.2)}}{\Gamma \vdash K <_c K''}\,\text{(SK.8)}$$

  On peut éliminer SK.5 de la prémisse gauche par hypothèse d'induction car la taille de d_1 est inférieure à s.

  On suppose maintenant que d_1 se termine par une des règles SK.2, SK.3 ou SK.4. Ces cas sont similaires. La différence se situe au niveau des règles SK.5 et SK.6. Voyons en détails la combinaison des régles SK.4, SK.5 et SK.4,SK.6.

- Cas $d \equiv$

$$\dfrac{\dfrac{\Gamma \overset{d'_1}{\vdash} K'_1 <_{c_1} K_1 \qquad \Gamma, x' : K'_1 \overset{d''_1}{\vdash} [c_1(x')/x]K_2 <_{c_2} K'_2 \qquad \Gamma, x : K_1 \overset{d'''_1}{\vdash} K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K'_1)K'_2}\,\text{(SK.4)} \qquad \Gamma \overset{d_2}{\vdash} (x' : K'_1)K'_2 = (x'' : K''_1)K''_2}{\Gamma \vdash (x : K_1)K_2 <_c (x'' : K''_1)K''_2}\,\text{(SK.5)}$$

  Le lemme 4.25 nous permet de définir les dérivations $\Gamma \overset{h_1}{\vdash} K'_1 = K''_1$ et $\Gamma, x' : K'_1 \overset{h_2}{\vdash} K'_2 = K''_2$; avec $h_1 \equiv \mathbf{P}_=(d_2)$ et $h_2 \equiv \mathbf{P}'_=(d_2)$. On obtient la dérivation suivante :

$$
\cfrac{\cfrac{\Gamma \overset{d_1'}{\vdash} K_1' <_{c_1} K_1 \qquad \Gamma \overset{h_1}{\vdash} K_1' = K_1''}{\Gamma \vdash K_1'' <_{c_1} K_1}\ \text{(SK.5)} \qquad \cfrac{\Gamma, x' : K_1' \overset{d_1''}{\vdash} [c_1(x')/x]K_2 <_{c_2} K_2' \qquad \Gamma, x' : K_1' \overset{h_2}{\vdash} K_2' = K_2''}{\Gamma \vdash K <_c K'}\ \text{(SK.5)} \qquad \Gamma, x : K_1 \overset{d_1'''}{\vdash} K_2 \textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x'' : K_1'')K_2''}\ \text{(SK.4)}
$$

On peut appliquer ici l'hypothèse d'induction pour éliminer la règle SK.5 des prémisses de gauche et du centre. En effet, le paramètre r diminue.

- Cas $d \equiv$

$$
\cfrac{\cfrac{\Gamma \overset{d_1'}{\vdash} K_1' <_{c_1} K_1 \qquad \Gamma, x' : K_1' \overset{d_1''}{\vdash} [c_1(x')/x]K_2 <_{c_2} K_2' \qquad \Gamma, x : K_1 \overset{d_1'''}{\vdash} K_2 \textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K_1')K_2'}\ \text{(SK.4)} \qquad \Gamma \overset{d_2}{\vdash} (x : K_1)K_2 = (x'' : K_1'')K_2''}{\Gamma \vdash (x'' : K_1'')K_2'' <_c (x' : K_1')K_2'}\ \text{(SK.6)}
$$

Le lemme 4.25 nous permet de définir les dérivations $\Gamma \overset{h_1}{\vdash} K_1 = K_1''$ et $\Gamma, x : K_1 \overset{h_2}{\vdash} K_2 = K_2''$, avec $h_1 \equiv \mathbf{P}_=(d_2)$ et $h_2 \equiv \mathbf{P}'_=(d_2)$.
Le lemme 4.16 nous permet d'obtenir les dérivations $\Gamma \overset{g_1}{\vdash} K_1' \textbf{kind}$ et $\Gamma \overset{g_2}{\vdash} c_1 : (K_1')K_1$, avec $g_1 \equiv \mathbf{l}^\circ_<(d_1')$ et $g_2 \equiv \mathbf{co}^\circ(d_1')$.

Considérons maintenant la dérivation : $g_1' \equiv \dfrac{\Gamma \overset{g_1}{\vdash} K_1' \textbf{kind}}{\Gamma, x' : K_1' \vdash \textbf{valid}}\ (1.2)$

Nous pouvons définir les dérivations suivantes (lemme 4.19) :

$$
h_2' \equiv \mathbf{E}^\circ\left(\frac{\Gamma, x : K_1 \overset{h_2}{\vdash} K_2 = K_2'' \qquad \Gamma, x' : K_1' \overset{g_1'}{\vdash} \textbf{valid}}{\Gamma, x' : K_1', x : K_1 \vdash K_2 = K_2''}\ \text{(wkn)} \right)
$$

ainsi que $g_2' \equiv$

$$
\mathbf{E}^\circ\left(\cfrac{\cfrac{\Gamma \overset{g_2}{\vdash} c_1 : (K_1')K_1 \qquad \Gamma, x' : K_1' \overset{g_1'}{\vdash} \textbf{valid}}{\Gamma, x' : K_1' \vdash c_1 : (K_1')K_1}\ \text{(wkn)} \qquad \cfrac{\Gamma, x' : K_1' \overset{g_1'}{\vdash} \textbf{valid}}{\Gamma, x' : K_1' \vdash x' : K_1'}\ (1.3)}{\Gamma, x' : K_1' \vdash c_1(x') : K_1}\ (5.5) \right)
$$

Prenons maintenant la dérivation suivante, dans laquelle la substitution est éliminée :

$$
h_2'' \equiv \mathbf{E}^\circ\left(\frac{\Gamma, x' : K_1', x : K_1 \overset{h_2'}{\vdash} K_2 = K_2'' \qquad \Gamma, x' : K_1' \overset{g_2'}{\vdash} c_1(x') : K_1}{\Gamma, x' : K_1' \vdash [c_1(x')/x]K_2 = [c_1(x')/x]K_2''}\ (6.4) \right)
$$

Finalement on obtient la dérivation :

$$\dfrac{\dfrac{\Gamma \overset{d'_1}{\vdash} K'_1 <_{c_1} K_1 \qquad \Gamma \overset{h_1}{\vdash} K_1 = K''_1}{\Gamma \vdash K'_1 <_{c_1} K''_1}\text{(SK.5)} \qquad \dfrac{\Gamma, x' : K'_1 \overset{d''_1}{\vdash} [c_1(x')/x]K_2 <_{c_2} K'_2 \qquad \Gamma, x' : K'_1 \overset{h''_2}{\vdash} [c_1(x')/x]K_2 = [c_1(x')/x]K''_2}{\Gamma, x' : K'_1 \vdash [c_1(x')/x]K''_2 <_{c_2} K'_2}\text{(SK.6)} \qquad \Gamma, x : K_1 \overset{d'''_1}{\vdash} K''_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x'' : K''_1)K''_2}\text{(SK.4)}$$

On peut appliquer ici l'hypothèse d'induction pour éliminer la règle SK.5 des prémisses de gauche et du centre. En effet le paramètre r diminue.

$\square$

Lemme 4.35. *Il existe un algorithme qui transforme toute $UTT^r[R]^-_{ok}$-dérivation d de la forme* $$\dfrac{\Gamma \overset{d_1}{\vdash} K <_c K' \qquad \Gamma \overset{d_2}{\vdash} K' <_{c'} K''}{\Gamma \vdash K <_{[x:K]c'(c(x))} K''}\text{(SK.7)}$$

où d_1 et d_2 ne contiennent pas les règles SK.5, SK.6 et SK.7, en une dérivation d' du jugement $\Gamma \vdash K <_{c^*} K''$. De plus d' ne contient pas les règles SK.5, SK.6 et SK.7 et le jugement $\Gamma \vdash [x : K]c'(c(x)) = c^*$ est dérivable dans UTT^r.

Démonstration. Grâce au lemme 4.33, on sait que $rank(K) = rank(K') = rank(K'') = r$. la preuve se fait par induction sur r.

Cas de base. Quand $r = 0$, $K \equiv El(A)$, $K' \equiv El(A')$ et $K'' \equiv El(A'')$. On peut conclure à l'aide de l'algorithme El^- (lemme 4.28) et de la règle ST.4. Il est facilement prouvable que si une dérivation h ne contient les règles SK.5, SK.6 et SK.7, alors $El^-(h)$ ne les contient pas non plus.

Pas d'induction. Si $r < 0$ alors $K \equiv (x : K_1)K_2$, $K' \equiv (x : K'_1)K'_2$ et $K'' \equiv (x : K''_1)K''_2$. Comme les dérivations des prémisses ne contiennent pas les règles de transitivité pour les sous-sortes, alors les dernières règles pour d_1 et d_2 ne peuvent être qu'une des règles SK.2, SK.3 et SK.4.

Prenons comme exemple le cas SK.4/SK.4

$$d_1 \equiv \dfrac{\Gamma \overset{e_1}{\vdash} K'_1 <_{c_1} K_1 \qquad \Gamma, x' : K'_1 \overset{e_2}{\vdash} [c_1(x')/x]K_2 <_{c_2} K'_2 \qquad \Gamma, x : K_1 \overset{e_3}{\vdash} K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K'_1)K'_2}\text{(SK.4)}$$

où $c \equiv [f : (x : K_1)K_2][x' : K'_1]c_2(f(c_1(x')))$.

$$d_2 \equiv \frac{\Gamma \overset{e'_1}{\vdash} K''_1 <_{c'_1} K'_1 \qquad \Gamma, x'' : K''_1 \overset{e'_2}{\vdash} [c'_1(x'')/x']K'_2 <_{c'_2} K''_2 \qquad \Gamma, x' : K'_1 \overset{e'_3}{\vdash} K'_2\textbf{kind}}{\Gamma \vdash (x' : K'_1)K'_2 <_{c'} (x'' : K''_1)K''_2} \text{(SK.4)}$$

où $c' \equiv [f' : (x' : K'_1)K'_2][x'' : K''_1]c'_2(f'(c'_1(x'')))$.

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} (x : K_1)K_2 <_c (x' : K'_1)K'_2 \qquad \Gamma \overset{d_2}{\vdash} (x' : K'_1)K'_2 <_{c'} (x'' : K''_1)K''_2}{\Gamma \vdash (x : K_1)K_2 <_{[v:(x:K_1)K_2]c'(c(v))} (x'' : K''_1)K''_2} \text{(SK.7)}$$

Le terme de coercion de la conclusion de d sera :
(1) $[v : (x : K_1)K_2]c'(c(v)) \equiv [v : (x : K_1)K_2][f' : (x' : K'_1)K'_2][x'' : K''_1]c'_2(f'(c'_1(x'')))([f : (x : K_1)K_2][x' : K'_1]c_2(f(c_1(x')))(v))$.

La dérivation sans transitivité que nous recherchons aura forcément la forme :

$$\frac{\Gamma \overset{?_1}{\vdash} K''_1 <_{c^*_1} K_1 \qquad \Gamma, x'' : K''_1 \overset{?_2}{\vdash} [c^*_1(x'')/x]K_2 <_{c^*_2} K''_2 \qquad \Gamma, x : K_1 \overset{?_3}{\vdash} K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_{c^*} (x'' : K''_1)K''_2} \text{(SK.4)}$$

Pour $?_3$, nous prenons la dérivation e_3, et pour $?_1$ on peut prendre

$$g_0 \equiv \frac{\Gamma \overset{e'_1}{\vdash} K''_1 <_{c'_1} K'_1 \qquad \Gamma \overset{e_1}{\vdash} K'_1 <_{c_1} K_1}{\Gamma \vdash K''_1 <_{[z'':K''_1]c_1(c'_1(z))} K_1} \text{(SK.7)}$$

où SK.7 est éliminée par hypothèse d'induction(r diminue).

On cherche maintenant $?_2$.
Notons que $c^*_1 \equiv [z'' : K''_1]c_1(c'_1(z))$ et que $[z'' : K''_1]c_1(c'_1(z''))(x'') \overset{\beta}{=} c_1(c'_1(x''))$.
Cherchons une dérivation de $\Gamma, x'' : K''_1 \overset{?_2}{\vdash} [c_1(c'_1(x''))/x]K_2 <_{c^*_2} K''_2$.

Grâce au lemme 4.24, nous sommes en mesure de construire les dérivations :
$\Gamma \overset{h_1}{\vdash} K''_1\textbf{kind}$ et $\Gamma \overset{h_2}{\vdash} c'_1 : (K''_1)K'_1$ à partir de e'_1, ainsi que
$\Gamma \overset{h_3}{\vdash} c_1 : (K'_1)K_1$ à partir de e_1.

Considérons maintenant les dérivations suivantes :

$$g_1 \equiv \frac{\Gamma \overset{h_1}{\vdash} K_1''\textbf{kind}}{\Gamma, x'' : K_1'' \vdash \textbf{valid}}\ (1.2)$$

$$g_2 \equiv \frac{\Gamma \overset{h_1}{\vdash} K_1''\textbf{kind}}{\Gamma, z'' : K_1'' \vdash \textbf{valid}}\ (1.2)$$

$$g_3 \equiv \mathbf{E}^\circ\Big(\frac{\Gamma \overset{h_3}{\vdash} c_1 : (K_1')K_1 \qquad \Gamma, x'' : K_1'' \overset{g_1}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'' \vdash c_1 : (K_1')K_1}\ (\text{wkn})\Big)$$

$$g_4 \equiv \mathbf{E}^\circ\Big(\frac{\Gamma \overset{h_2}{\vdash} c_1' : (K_1'')K_1' \qquad \Gamma, x'' : K_1'' \overset{g_1}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'' \vdash c_1' : (K_1'')K_1'}\ (\text{wkn})\Big)$$

$$g_5 \equiv \frac{\Gamma, x'' : K_1'' \overset{g_1}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'' \vdash x'' : K_1''}\ (1.3)$$

$$g_6 \equiv \dfrac{\Gamma, x'' : K_1'' \overset{g_3}{\vdash} c_1 : (K_1')K_1 \qquad \dfrac{\Gamma, x'' : K_1'' \overset{g_4}{\vdash} c_1' : (K_1'')K_1' \qquad \Gamma, x'' : K_1'' \overset{g_5}{\vdash} x'' : K_1''}{\Gamma, x'' : K_1'' \vdash c_1'(x'') : K_1'}\ (5.5)}{\Gamma, x'' : K_1'' \vdash c_1(c_1'(x'')) : K_1}\ (5.5)$$

$$g_7 \equiv \mathbf{E}^\circ\Big(\frac{\Gamma, z'' : K_1'' \overset{g_2}{\vdash} \textbf{valid} \qquad \Gamma, x'' : K_1'' \overset{g_1}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'', z'' : K_1'' \vdash \textbf{valid}}\ (\text{wkn})\Big)$$

$$g_8 \equiv \mathbf{E}^\circ\Big(\frac{\Gamma \overset{h_3}{\vdash} c_1 : (K_1')K_1 \qquad \Gamma, x'' : K_1'', z'' : K_1'' \overset{g_7}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'', z'' : K_1'' \vdash c_1 : (K_1')K_1}\ (\text{wkn})\Big)$$

$$g_9 \equiv \mathbf{E}^\circ\Big(\frac{\Gamma \overset{h_2}{\vdash} c_1' : (K_1'')K_1' \qquad \Gamma, x'' : K_1'', z'' : K_1'' \overset{g_7}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'', z'' : K_1'' \vdash c_1' : (K_1'')K_1'}\ (\text{wkn})\Big)$$

$$g_{10} \equiv \frac{\Gamma, x'' : K_1'', z'' : K_1'' \overset{g_7}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'', z'' : K_1'' \vdash z'' : K_1''}\ (1.3)$$

$$g_{11} \equiv \frac{\Gamma, x'' : K_1'', z'' : K_1'' \overset{g_8}{\vdash} c_1 : (K_1')K_1 \qquad \dfrac{\Gamma, x'' : K_1'', z'' : K_1'' \overset{g_9}{\vdash} c_1' : (K_1'')K_1' \qquad \Gamma, x'' : K_1'', z'' : K_1'' \overset{g_{10}}{\vdash} z'' : K_1''}{\Gamma, x'' : K_1'', z'' : K_1'' \vdash c_1'(z'') : K_1'} \text{(5.5)}}{\Gamma, x'' : K_1'', z'' : K_1'' \vdash c_1(c_1'(z'')) : K_1} \text{(5.5)}$$

$$g_{12} \equiv \frac{\Gamma, x'' : K_1'', z'' : K_1'' \overset{g_{11}}{\vdash} c_1(c_1'(z'')) : K_1 \qquad \Gamma, x'' : K_1'' \overset{g_5}{\vdash} x'' : K_1''}{\Gamma, x'' : K_1'' \vdash [z'' : K_1''](c_1(c_1'(z'')))(x'') = c_1(c_1'(x'')) : K_1} \text{(5.7)}$$

$$g_{13} \equiv \mathbf{E}(\frac{\Gamma, x : K_1 \overset{e_3}{\vdash} K_2\textbf{kind} \qquad \Gamma, x'' : K_1'' \overset{g_1}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'', x : K_1 \vdash K_2\textbf{kind}} \text{(wkn))}$$

$$g_{14} \equiv \mathbf{E}(\frac{\Gamma, x'' : K_1'', x : K_1 \overset{g_{13}}{\vdash} K_2\textbf{kind} \qquad \Gamma, x'' : K_1'' \overset{g_{12}}{\vdash} [z'' : K_1''](c_1(c_1'(z'')))(x'') = c_1(c_1'(x'')) : K_1}{\Gamma, x'' : K_1'' \vdash [[z'' : K_1''](c_1(c_1'(z'')))(x'')/x]K_2 = [c_1(c_1'(x''))/x]K_2} \text{(6.6))}$$

$$g_{15} \equiv \mathbf{E}(\frac{\mathbf{E}(\dfrac{\Gamma, x' : K_1' \overset{e_2}{\vdash} [c_1(x')/x]K_2 <_{c_2} K_2' \qquad \Gamma, x'' : K_1'' \overset{g_1}{\vdash} \textbf{valid}}{\Gamma, x'' : K_1'', x' : K_1' \vdash [c_1(x')/x]K_2 <_{c_2} K_2'} \text{(wkn))} \qquad \dfrac{\Gamma, x'' : K_1'' \overset{g_4}{\vdash} c_1' : (K_1'')K_1' \qquad \Gamma, x'' : K_1'' \overset{g_5}{\vdash} x'' : K_1''}{\Gamma, x'' : K_1'' \vdash c_1'(x'') : K_1'} \text{(5.5)}}{\Gamma, x'' : K_1'' \vdash [c_1(c_1'(x''))/x]K_2 <_{[c_1'(x'')/x']c_2} K_2'} \text{(SK.9))}$$

Pour $?_2$ on peut donc prendre :

$$g_{16} \equiv \frac{\Gamma, x'' : K_1'' \overset{g_{15}}{\vdash} [c_1(c_1'(x''))/x]K_2 <_{[c_1'(x'')/x']c_2} K_2' \qquad \Gamma, x'' : K_1'' \overset{e_2'}{\vdash} [c_1'(x'')/x']K_2' <_{c_2'} K_2''}{\Gamma, x'' : K_1'' \vdash [c_1(c_1'(x''))/x]K_2 <_{c_2^*} K_2''} \text{(SK.7)}$$

Les substitutions ne changent pas le rang et on élimine SK.7 par hypothèse d'induction. De plus on peut dire que $c_2^* \equiv [u : [c_1(c_1'(x''))/x]K_2]c_2'([c_1'(x'')/x']c_2(u))$.
On obtient finalement :

$$\frac{\Gamma \overset{g_0}{\vdash} K_1'' <_{c_1^*} K_1 \qquad \Gamma, x'' : K_1'' \overset{g_{16}}{\vdash} [c_1^*(x'')/x]K_2 <_{c_2^*} K_2'' \qquad \Gamma, x : K_1 \overset{e_3}{\vdash} K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_{c^*} (x'' : K_1'')K_2''} \text{(SK.4)}$$

Connaissant la correspondance entre c^*, c_1^* et c_2^* on a :
$c^* \equiv [v : (x : K_1)K_2](c_2^*(v(c_1^*(x''))))$. Or nous rappelons que
$c_1^* \equiv [z'' : K_1'']c_1(c_1'(z))$.
On a donc :
(2) $c^* \equiv [v : (x : K_1)K_2][x'' : K_1'']([u : [c_1(c_1'(x''))/x]K_2]c_2'([c_1'(x'')/x']c_2(u))(v([z'' : K_1'']c_1(c_1'(z))(x''))))$.
Il nous faut comparer ce terme avec le terme de départ (1), qui est :

(1) $[v : (x : K_1)K_2]([f' : (x' : K'_1)K'_2][x'' : K''_1]c'_2(f'(c'_1(x'')))[f : (x : K_1)K_2][x' : K'_1]c_2(f(c_1(x')))(v))$.

Nous laissons le lecteur dériver le jugement représentant cette égalité. Nous allons juste vérifier quelles conversions doivent être appliquées et dans quel ordre. Dans c^* on convertit $[z'' : K''_1]c_1(c'_1(z''))(x'') \rhd_\beta c_1(c'_1(x''))$. Le terme $v(c_1(c'_1(x'')))$ est de la sorte $[c_1(c'_1(x''))/x]K_2$. La prochaine β-conversion utilise la variable liée u, et le résultat est :
$[v : (x : K_1)K_2][x'' : K''_1](c'_2([c'_1(x'')/x']c_2(v(c_1(c'_1(x'')))))).$

Dans l'autre terme (2), on applique d'abord la β-réduction avec la variable liée f, on a :
$[f : (x : K_1)K_2][x' : K'_1]c_2(f(c_1(x')))(v) \rhd_\beta [x' : K'_1]c_2(v(c_1(x'))).$

Le terme $[x' : K'_1]c_2(v(c_1(x')))$ est de la sorte $(x' : K'_1)K'_2$. On applique ensuite une β-réduction avec la variable liée f', on a :
$[f' : (x' : K'_1)K'_2][x'' : K''_1](c'_2(f'(c'_1(x''))))[x' : K'_1](c_2(v(c_1(x')))) \rhd_\beta$
$[x'' : K''_1](c'_2([x' : K'_1](c_2(v(c_1(x'))))(c'_1(x'')))).$

Puis on applique une β-réduction avec la variable liée x', on a :
$[x'' : K''_1](c'_2([x' : K'_1](c_2(v(c_1(x'))))(c'_1(x'')))) \rhd_\beta [x'' : K''_1](c'_2([c'_1(x'')/x']c_2(v(c_1(c'_1(x'')))))).$

On a considéré ici le fait que x' n'apparaît ni dans c_1, ni dans v.
Le terme (1) est bien converti en
$[v : (x : K_1)K_2][x'' : K''_1](c'_2([c'_1(x'')/x']c_2(v(c_1(c'_1(x'')))))).$

Ce qui correspond bien au terme obtenu à partir de c^*.

La démonstration pour les cas SK.4/SK.3, SK.4/SK.2, SK.3/SK.2, SK.2/SK.2 est essentiellement la même.

$\square$

Theorème 4.36 (Elimination de la transitivité pour les sous-sortes dans $UTT^r[R]^-_{ok}$**).**
Il existe un algorithme qui transforme toute dérivation du jugement $\Gamma \vdash K <_c K'$ *dans* $UTT^r[R]^-_{ok}$ *en une dérivation du jugement* $\Gamma \vdash K <_{c'} K'$ *dans le même calcul mais ne contenant pas les règles SK.5, SK.6 et SK.7, telle que* $\Gamma \vdash c' = c : (K)K'$ *soit dérivable dans* UTT^r.

Démonstration. Par induction sur le nombre de règles de transitivité r. On utilise les lemmes 4.35 et 4.34.

□

Corollaire 4.37. *Si les conditions de cohérence sont satisfaites pour le sous-typage alors dans $UTT^r[R]_{ok}$, on a :*

a) si $\Gamma \vdash K = K'$ alors $\Gamma \vdash K <_c K'$ n'est pas dérivable ;

b) si $\Gamma \vdash K <_c K'$ et $\Gamma \vdash K <_{c'} K'$ sont dérivables, alors $\Gamma \vdash c' = c : (K)K'$.

Démonstration. Nous savons que $rank(K) = rank(K') = r$ (lemme 4.33).

Montrons a). Supposons qu'il existe une dérivation d de $\Gamma \vdash K <_c K'$ de taille s. La preuve se fait par induction sur $r\omega + s$.

Cas de base. Quand r=0, on a $K \equiv El(A)$ et $K' \equiv El(A')$. Par hypothèse le jugement $\Gamma \vdash El(A) = El(A')$ est dérivable, dans ce cas nous pouvons dériver $\Gamma \vdash A = A'$: **Type**. De plus la dérivabilité de $\Gamma \vdash El(A) <_c El(A')$ implique celle de $\Gamma \vdash A <_c A'$: **Type** (lemme 4.28). Des deux jugements $\Gamma \vdash A = A'$: **Type** et $\Gamma \vdash A <_c A'$: **Type** nous pouvons obtenir la dérivation de $\Gamma \vdash A' <_c A'$: **Type** de la manière suivante :

$$\frac{\Gamma \vdash A <_c A' : \textbf{Type} \qquad \Gamma \vdash A = A' : \textbf{Type}}{\Gamma \vdash A' <_c A' : \textbf{Type}} \text{ (ST.2)}$$

Ce qui contredit les conditions de cohérence pour le sous-typage.

Pas d'induction. Soit $r > 0$. $K \equiv (x : K_1)K_2$ et $K' \equiv (x : K_1')K_2'$. Le théorème 4.36 nous permet de considérer que d est une dérivation sans règle de transitivité. Si la dernière règle de d est SK.8, on considère la première prémisse. Le rang n'est pas changé mais la taille de la dérivation a diminué. On peut donc appliquer l'hypothèse d'induction et obtenir une contradiction.
Si d se termine par SK.2 ou SK.4, la premiere prémisse est $\Gamma \vdash K_1 <_{c_1} K_1'$, or le lemme 4.25 nous permet d'obtenir $\Gamma \overset{\mathbf{P}_{=}(d)}{\vdash} K_1 = K_1'$. Ce qui est en contradiction avec l'hypothèse d'induction (que nous pouvons appliquer car $r\omega + s$ a diminué).

Si d se termine par SK.3, la seconde prémisse est $\Gamma, x : K_1' \vdash K_1 <_{c_2} K_1'$. Grâce au lemme 4.25 nous avons $\Gamma, x : K_1' \overset{\mathbf{P}''_{=}(d)}{\vdash} K_1 = K_1'$. Ce qui est en contradiction avec l'hypothèse d'induction (que nous pouvons appliquer car $r\omega + s$ à diminué).

Montrons b)
Nous considérons les dérivations d et d' de $\Gamma \vdash K <_c K'$ et $\Gamma \vdash K <_{c'} K'$ respectivement. Nous présumons grâce au théorème 4.36 que d et d' ne contiennent pas de règle de transitivité. Soit s et s' les tailles de d et d'. La preuve se fait par induction sur $\omega r + s + s'$. Les cas à considérer sont les couples possibles formés des règles SK.1, SK.2, SK.3, SK.4, SK.8.

Cas de base. Quand $r = 0$. $K \equiv El(A)$ et $K' \equiv El(A')$. Le lemme 4.28 nous permet d'affirmer que les jugements $\Gamma \overset{d}{\vdash} El(A) <_c El(A')$ et $\Gamma \overset{d'}{\vdash} El(A) <_{c'} El(A')$ sont obtenus par application de la règle SK.1 aux jugements de sous-typage $\Gamma \vdash A <_c A' :$ **Type** et $\Gamma \vdash A <_{c'} A' :$ **Type**. Nous avons $\Gamma \vdash c' = c : (K)K'$ par rapport aux conditions de cohérence pour le sous-typage.

Pas d'induction. Soit $r > 0$. $K \equiv (x : K_1)K_2$ et $K' \equiv (x : K_1')K_2'$.

- Prenons le cas où au moins une des deux dérivations d ou d' se termine par la règle de congruence SK.8. Supposons que ce soit d.

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} (x : K_1)K_2 <_{c_1} (x : K_1')K_2' \qquad \Gamma \overset{d_2}{\vdash} c_1 = c : ((x : K_1)K_2)(x : K_1')K_2'}{\Gamma \vdash (x : K_1)K_2 <_c (x : K_1')K_2'} \text{ (SK.8)}$$

  Par hypothèse d'induction nous avons le jugement $\Gamma \overset{d_2}{\vdash} c_1 = c' : ((x : K_1)K_2)(x : K_1')K_2'$. En effet s diminue tandis que r et s' restent inchangées. Nous pouvons donc dériver le jugement suivant :

$$\frac{\dfrac{\Gamma \overset{d_2}{\vdash} c_1 = c : ((x : K_1)K_2)(x : K_1')K_2'}{\Gamma \vdash c = c_1 : ((x : K_1)K_2)(x : K_1')K_2'} \text{ (2.5)} \qquad \Gamma \vdash c_1 = c' : ((x : K_1)K_2)(x : K_1')K_2'}{\Gamma \vdash c = c' : ((x : K_1)K_2)(x : K_1')K_2'} \text{ (2.6)}$$

- Prenons le cas où les dérivations d et d' se terminent toutes les deux par une des règles d'introduction de produit (SK.2, SK.3 ou SK.4). Le a) du corollaire implique que les prémisses de gauche de ces règles sont $\Gamma \vdash$

$K_1' = K_1$ ou, $\Gamma \vdash K_1' <_{c_1} K_1$ et $\Gamma \vdash K_1' <_{c_1'} K_1$.
Si la prémisses de gauche de chacune de ces règles est $\Gamma \vdash K_1' = K_1$, alors d et d' se terminent par SK.3. Les secondes prémisses seront forcément $\Gamma, x : K_1' \vdash K_2 <_{c_2} K_2'$ et $\Gamma, x : K_1' \vdash K_2 <_{c_2'} K_2'$. Par hypothèse d'induction $\Gamma, x : K_1' \vdash c_2 = c_2' : (K_2)K_2'$ (r diminue) ; ce qui implique que $\Gamma \vdash c = c' : ((x : K_1)K_2)(x : K_1')K_2'$, car $c \equiv [f : (x : K_1)K_2][x' : K_1']c_2(f(x))$ et $c' \equiv [f : (x : K_1)K_2][x' : K_1']c_2'(f(x))$.
Si les prémisses de gauche de ces règles sont $\Gamma \vdash K_1' <_{c_1} K_1$ et $\Gamma \vdash K_1' <_{c_1'} K_1$, par hypothèse d'induction nous avons $\Gamma \vdash c_1 = c_1' : (K_1')K_1$, ce qui implique que $\Gamma, x' : K_1' \overset{h}{\vdash} [c_1(x')/x]K_2 = [c_1'(x')/x]K_2$. Le a) du corollaire implique que les secondes prémisses sont soit :
$\Gamma, x' : K_1' \vdash [c_1(x')/x]K_2 = K_2'$ et $\Gamma, x' : K_1' \vdash [c_1'(x')/x]K_2 = K_2'$;
soit :
$\Gamma, x' : K_1' \overset{d_2}{\vdash} [c_1(x')/x]K_2 <_{c_2} K_2'$ et $\Gamma, x' : K_1' \overset{d_2'}{\vdash} [c_1'(x')/x]K_2 <_{c_2'} K_2'$.
Autrement dit :

- soit d et d' se terminent par SK.2, dans ce cas nous savons déjà que $\Gamma \vdash c_1 = c_1' : (K_1')K_1$, on peut donc dériver $\Gamma \vdash c = c' : ((x : K_1)K_2)(x : K_1')K_2'$, car $c \equiv [f : (x : K_1)K_2][x' : K_1']f(c_1(x'))$ et $c' \equiv [f : (x : K_1)K_2][x' : K_1']f(c_1'(x'))$;

- soit d et d' se terminent par SK.4, dans ce cas en appliquant SK.6 nous obtenons (le théorème 4.36 nous permet d'éliminer SK.6) :

$$h_2 \equiv \frac{\Gamma, x' : K_1' \overset{d_2}{\vdash} [c_1(x')/x]K_2 <_{c_2} K_2' \qquad \Gamma, x' : K_1' \overset{h}{\vdash} [c_1(x')/x]K_2 = [c_1'(x')/x]K_2}{\Gamma, x' : K_1' \vdash [c_1'(x')/x]K_2 <_{c_2} K_2'} \text{(SK.6)}$$

  Nous pouvons appliquer l'hypothèse d'induction à
  $\Gamma, x' : K_1' \overset{d_2'}{\vdash} [c_1'(x')/x]K_2 <_{c_2'} K_2'$ et $\Gamma, x' : K_1' \overset{h_2}{\vdash} [c_1'(x')/x]K_2 <_{c_2} K_2'$, pour obtenir $\Gamma, x' : K_1' \vdash c_2 = c_2' : ([c_1'(x')/x]K_2)K_2'$.
  Des jugements $\Gamma \vdash c_1 = c_1' : (K_1')K_1$ et $\Gamma, x' : K_1' \vdash c_2 = c_2' : ([c_1'(x')/x]K_2)K_2'$, nous pouvons dériver
  $\Gamma \vdash c = c' : ((x : K_1)K_2)(x : K_1')K_2'$, sachant que
  $c \equiv [f : (x : K_1)K_2][x' : K_1']c_2(f(c_1(x')))$ et $c' \equiv [f : (x : K_1)K_2][x' : K_1']c_2'(f(c_1'(x')))$.

$\square$

4.2.4 Complétion des coercions

Nous définissons dans cette section l'algorithme de complétion des coercions Ψ ainsi que ses relations avec les dérivations.

Définition 4.38. *Considérons une dérivation d d'un jugement $\Gamma \vdash J$, où $\Psi(d)$ est définie. $\Psi(d)$ détermine également une unique transformation syntaxique de tous les constituants de $\Gamma \vdash J$. Soit $\Psi_d(\Gamma \vdash J)$, le jugement final de $\Psi(d)$ et $\Gamma' \vdash J'$ un jugement quelconque (pas forcément dérivable) où Γ' est obtenu en typant les variables libres par les sorte-constituants de Γ, et J' est une des formes possibles de jugement construit à partir des constituants de $\Gamma \vdash J$.*
On dénote par $\Psi_d(\Gamma' \vdash J')$ le jugement obtenu en remplaçant chaque constituant de $\Gamma \vdash J$ utilisé dans $\Gamma' \vdash J'$ par le constituant correspondant dans $\Psi_d(\Gamma \vdash J)$.

Remarque 4.39.

- *Pour tout constituant c du jugement $\Gamma' \vdash J'$ vu ci-dessus, $\Psi_d(c)$ représente le constituant correspond à c du jugement $\Psi_d(\Gamma' \vdash J')$.*
- *Soit d_0 une sous-dérivation de d, du jugement $\Gamma_0 \vdash J_0$, on peut remplacer dans le jugement $\Psi_d(\Gamma \vdash J)$ chaque constituant c commun à $\Gamma \vdash J$ et $\Gamma_0 \vdash J_0$, par $\Psi_{d_0}(c)$.*
- *Soit une dérivation d' telle que $\Psi(d')$ est définie et $\Psi(d') \sim \Psi(d)$, alors quelque soit le jugement $\Gamma' \vdash J'$ on a $\Psi_d(\Gamma' \vdash J') = \Psi_{d'}(\Gamma' \vdash J')$.*
- *d étant une dérivation du jugement $\Gamma \vdash J$ alors $\Psi_d(\Gamma \vdash J)$ est le jugement final de $\Psi(d)$.*

Pour rappel, les constituants d'un jugement sont introduits à la définition 4.5.

4.2.4.1 Définition de la transformation Ψ

Afin d'utiliser les résultats de la section 4.2.3, nous définissons la transformation Ψ sur la classe des dérivations incluant les substitutions, la règle d'affaiblissement wkn, la règle de retypage de contexte 3.3 ainsi que les règles modifiées avec des prémisses supplémentaires. En principe, la modification d'un jugement dépend de sa dérivation : si nous considérons deux dérivations du même jugement, après l'application de Ψ à ces deux jugements nous pouvons obtenir deux dérivations dans UTT^r de deux jugements différents.

Ψ est définie par induction structurelle.

- Pour toute dérivation d dans $UTT^r[R]_{ok}$, on a $\Psi(d) \equiv d$.

- Pour chaque règle r n'ayant qu'une prémisse , si Ψ est définie pour la dérivation d_0 de la prémisse, alors la règle r doit être appliquée à la conclusion de $\Psi(d_0)$. Ce cas couvre les règles 1.2, 1.3, 2.1, 2.2, 2.4, 2.5, 4.1-4.3, 5.1, 5.3, 5.8, $SK.1$, T-$prop$, T-prf, ainsi que l'introduction des constantes. Dans le cas particulier des règles C-M, C-i, C-E, nous devons être en mesure de vérifier que la sorte Θ après insertion des coercions reste un préschéma inductif. Prenons par exemple le cas de la règle C-M :

  $$d \equiv \frac{PSCH_X(\bar{\Theta}) \qquad \Gamma, X : \textbf{Type} \overset{d_0}{\vdash} \Theta_i \textbf{kind}}{\Gamma \vdash M^X[\bar{\Theta}] : \textbf{Type}} \text{ (C-M)} \quad 1 \leq i \leq n.$$

  Nous avons $$\Psi(d) \equiv \frac{PSCH_X(\bar{\Theta}_0) \qquad \Gamma_0, X_0 : \textbf{Type} \overset{\Psi(d_0)}{\vdash} \Theta_{0i} \textbf{kind}}{\Gamma_0 \vdash M^X[\bar{\Theta_0}] : \textbf{Type}} \text{ (C-M)}$$
  $1 \leq i \leq n.$

  $\Psi(d)$ est définie si nous pouvons vérifier que $PSCH_X(\bar{\Theta}_0)$.

Pour les autres règles, nous présumons que Ψ est déjà définie pour les dérivations des prémisses. Dans ce cas, les constituants des prémisses transformées ne correspondent pas forcément. Pour rendre possible l'application de la même règle que dans la dérivation d'origine d, nous devons insérer certaines applications des règles de retypage (3.1-3.3). La dérivation $\Psi(d)$ est définie seulement s'il existe des UTT^r-dérivations de certains jugements d'égalité nécessaires. Les cas spéciaux sont ceux des règles coercitives $CA.1$, $CA.2$ et CD. Elles sont remplacées par d'autres règles permettant d'insérer les termes de coercion manquants. L'exemple 4.2 nous fournit une idée du fonctionnement de la transformation.
Les règles à considérer sont les suivantes (incluant à la fois la formulation affaiblie et la formulation normale pour les règles ayant deux formulations) : 2.3, 2.6, 3.1, 3.2, 3.3, 3.3', wkn, 5.2, 5.4-5.7, 6.1-6.7, 6.6', 6.7', ST.1-ST.5, ST.2', ST.3', SK.2-SK.9, SK.5', SK.6', CA.1, CA.2, CD, ϑ-eq, E-eq.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} k = k' : K \qquad \Gamma \overset{d_2}{\vdash} k' = k'' : K}{\Gamma \vdash k = k'' : K}\,(2.6)$

  Ce cas est similaire à celui des règles 2.3, 3.1, 3.2.
  Nous présumons que $\Psi(d_1)$ et $\Psi(d_2)$ sont définies. Nous ajoutons les indices 1 et 2 aux contextes, termes et sortes afin d'indiquer si ces derniers proviennent de la dérivation $\Psi(d_1)$ ou $\Psi(d_2)$.

$$\Psi(d) \equiv \dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} k_1 = k_1' : K_1 \qquad \dfrac{\Gamma_1 \overset{?_1}{\vdash} k_1' = k_2' : K_1 \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2' = k_2'' : K_2 \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2' = k_2'' : K_2}\,(3.3) \qquad \Gamma_1 \overset{?_3}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2' = k_2'' : K_1}\,(3.2)}{\Gamma_1 \vdash k_1' = k_2'' : K_1}\,(2.6)}{\Gamma_1 \vdash k_1 = k_2'' : K_1}\,(2.6)$$

  Remarquons que nous ajustons la prémisse de droite afin de correspondre à celle de gauche par application de règles de retypage appropriées. Toutes les égalités dénotées par les points d'interrogation peuvent être dérivées si nous pouvons prouver dans UTT^r l'égalité de jugements suivante :
  $\Psi_{d_1}(\Gamma \vdash k' : K) = \Psi_{d_2}(\Gamma \vdash k' : K)$.

- Cas $d \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} J \qquad \Gamma \overset{d_2}{\vdash} K = K'}{\Gamma, x : K', \Gamma' \vdash J}\,(3.3)$

$$\Psi(d) \equiv \dfrac{\Gamma_1, x : K_1, \Gamma_1' \overset{\Psi(d_1)}{\vdash} J_1 \qquad \dfrac{\Gamma_1 \overset{?_1}{\vdash} K_1 = K_2 \qquad \dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} K_2 = K_2' \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash K_2 = K_2'}\,(3.3)}{\Gamma_1 \vdash K_1 = K_2'}\,(2.3)}{\Gamma_1, x : K_2', \Gamma_1' \vdash J_1}\,(3.3)$$

  Comme précédemment, les égalités dénotées par les points d'interrogation peuvent être dérivées si nous pouvons prouver dans UTT^r l'égalité de jugements suivante : $\Psi_{d_1}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K\mathbf{kind})$.

- Cas $d \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} J \qquad \Gamma \overset{d_2}{\vdash} K = K' \qquad \Gamma \overset{d_2}{\vdash} K'\mathbf{kind}}{\Gamma, x : K', \Gamma' \vdash J}\,(3.3')$

Ce cas montre quelle est la différence quand on rajoute des prémisses.

$$\Psi(d) \equiv \dfrac{\Gamma_1, x : K_1, \Gamma_1' \overset{\Psi(d_1)}{\vdash} J_1 \qquad \dfrac{\dfrac{\Gamma_1 \overset{?_1}{\vdash} K_1 = K_2 \qquad \dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} K_2 = K_2' \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash K_2 = K_2'}\,(3.3)}{\Gamma_1 \vdash K_1 = K_2'}\,(2.3) \qquad \Gamma_1 \overset{?_3}{\vdash} K_2' = K_3'}{\Gamma_1 \vdash K_1 = K_3'}\,(2.3) \qquad \dfrac{\Gamma_3 \overset{\Psi(d_3)}{\vdash} K_3' \textbf{kind} \qquad \overset{?_4}{\vdash} \Gamma_3 = \Gamma_1}{\Gamma_1 \vdash K_3' \textbf{kind}}\,(3.3)}{\Gamma_1, x : K_3', \Gamma_1' \vdash J_1}\,(3.3')$$

Dans ce cas, les égalités dont on a besoin sont contenues dans $\Psi_{d_1}(\Gamma \vdash K\textbf{kind}) = \Psi_{d_2}(\Gamma \vdash K\textbf{kind})$, et $\Psi_{d_2}(\Gamma \vdash K'\textbf{kind}) = \Psi_{d_3}(\Gamma \vdash K'\textbf{kind})$.

- Cas $d \equiv \dfrac{\Gamma, \Gamma' \overset{d_1}{\vdash} J \qquad \Gamma, \Gamma'' \overset{d_2}{\vdash} \textbf{valid}}{\Gamma, \Gamma''\Gamma' \vdash J}\,(\text{wkn})$

  $$\Psi(d) \equiv \dfrac{\Gamma_1\Gamma_1' \overset{\Psi(d_1)}{\vdash} J_1 \qquad \dfrac{\Gamma_2, \Gamma_2'' \overset{\Psi(d_2)}{\vdash} \textbf{valid} \qquad \overset{?_1}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1, \Gamma_2'' \vdash \textbf{valid}}\,(3.3)}{\Gamma_1, \Gamma_2'', \Gamma_1' \vdash J_1}\,(\text{wkn})$$

  Les égalités dont on a besoin sont contenues dans $\Psi_{d_1}(\Gamma \vdash \textbf{valid}) = \Psi_{d_2}(\Gamma \vdash \textbf{valid})$.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f : (x : K)K' \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma \vdash f(k) : [k/x]K'}\,(5.5)$

  Ce cas est similaire à celui des règles 5.2, 5.6, 5.7.

  $$\Psi(d) \equiv \dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} f_1 : (x : K_1)K_1' \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2 : K_2 \qquad \overset{?_1}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2 : K_2}\,(3.3) \qquad \Gamma_1 \overset{?_2}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2 : K_1}\,(3.1)}{\Gamma_1 \vdash f_1(k_2) : [k_2/x]K_1'}\,(5.5)$$

  Les égalités dont on a besoin sont contenues dans $\Psi_{d_1}(\Gamma \vdash K\textbf{kind}) = \Psi_{d_2}(\Gamma \vdash K\textbf{kind})$.

- Cas $d \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} k' = k'' : K' \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]k' = [k/x]k'' : [k/x]K'}$ (6.5)

  Ce cas est similaire aux cas des substitutions simples 6.1-6.5.

$$\Psi(d) \equiv \dfrac{\Gamma_1, x : K_1, \Gamma'_1 \overset{\Psi(d_1)}{\vdash} k'_1 = k''_1 : K'_1 \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2 : K_2 \qquad \overset{?_1}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2 : K_2}\text{(3.3)} \qquad \Gamma_1 \overset{?_2}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2 : K_1}\text{(3.1)}}{\Gamma_1, [k_2/x]\Gamma'_1 \vdash [k_2/x]k'_1 = [k_2/x]k''_1 : [k_2/x]K'_1}\text{(6.5)}$$

  Les égalités dont on a besoin sont contenues dans
  $\Psi_{d_1}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K\mathbf{kind})$.

- Cas $d \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} q : Q \qquad \Gamma \overset{d_2}{\vdash} k = k' : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]q = [k/x]q : [k'/x]Q}$ (6.7)

  Le cas 6.6 est similaire.

$$\Psi(d) \equiv \dfrac{\Gamma_1, x : K_1, \Gamma'_1 \overset{\Psi(d_1)}{\vdash} q_1 : Q_1 \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2 = k'_2 : K_2 \qquad \overset{?_1}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2 = k'_2 : K_2}\text{(3.3)} \qquad \Gamma_1 \overset{?_2}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2 = k'_2 : K_1}\text{(3.2)}}{\Gamma_1, [k_2/x]\Gamma'_1 \vdash [k_2/x]q_1 = [k'_2/x]q_1 : [k_2/x]Q}\text{(6.7)}$$

  Les égalités dont on a besoin sont contenues dans
  $\Psi_{d_1}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K\mathbf{kind})$.

- Cas $d \equiv \dfrac{\Gamma, x : K, \Gamma' \overset{d_1}{\vdash} q : Q \qquad \Gamma \overset{d_2}{\vdash} k = k' : K \quad \Gamma \overset{d_3}{\vdash} k : K \quad \Gamma \overset{d_4}{\vdash} k' : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]q = [k/x]q : [k'/x]Q}$ (6.7')

  Considérons la dérivation $D_1 \equiv$

$$\dfrac{\dfrac{\Gamma_1 \overset{?_3}{\vdash} k_3 = k_2 : K_1 \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2 = k_2' : K_2 \qquad \overset{?_1}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2 = k_2' : K_2}\,(3.3) \qquad \Gamma_1 \overset{?_2}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2 = k_2' : K_1}\,(3.2)}{\Gamma_1 \vdash k_3 = k_2' : K_1}\,(2.6) \qquad \Gamma_1 \overset{?_4}{\vdash} k_2' = k_4' : K_1}{\Gamma_1 \vdash k_3 = k_4' : K_1}\,(2.6)$$

ainsi que $D_2 \equiv$

$$\dfrac{\dfrac{\Gamma_3 \overset{\Psi(d_3)}{\vdash} k_3 : K_3 \qquad \overset{?_5}{\vdash} \Gamma_3 = \Gamma_1}{\Gamma_1 \vdash k_3 : K_3}\,(3.3) \qquad \Gamma_1 \overset{?_6}{\vdash} K_3 = K_1}{\Gamma_1 \vdash k_3 : K_1}\,(3.1)$$

et $D_3 \equiv$

$$\dfrac{\dfrac{\Gamma_4 \overset{\Psi(d_4)}{\vdash} k_4' : K_4 \qquad \overset{?_7}{\vdash} \Gamma_4 = \Gamma_1}{\Gamma_1 \vdash k_4' : K_4}\,(3.3) \qquad \Gamma_1 \overset{?_8}{\vdash} K_4 = K_1}{\Gamma_1 \vdash k_4' : K_1}\,(3.1)$$

finalement on a

$$\Psi(d) \equiv \dfrac{\Gamma, x : K_1, \Gamma_1' \overset{\Psi(d_1)}{\vdash} q_1 : Q_1 \qquad \Gamma_1 \overset{D_1}{\vdash} k_3 = k_4' : K_1 \qquad \Gamma_1 \overset{D_2}{\vdash} k_3 : K_1 \quad \Gamma_1 \overset{D_3}{\vdash} k_4' : K_1}{\Gamma_1, [k_3/x]\Gamma_1' \vdash [k_3/x]q_1 = [k_4'/x]q_1 : [k_3/x]Q_1}\,(6.7')$$

Dans ce cas, les égalités dont on a besoin sont contenues dans $\Psi_{d_1}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K\mathbf{kind})$, $\Psi_{d_1}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash K\mathbf{kind})$, $\Psi_{d_1}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_4}(\Gamma \vdash K\mathbf{kind})$, $\Psi_{d_2}(\Gamma \vdash k : K) = \Psi_{d_3}(\Gamma \vdash k : K)$, $\Psi_{d_2}(\Gamma \vdash k' : K) = \Psi_{d_4}(\Gamma \vdash k' : K)$.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} A : \mathbf{Type} \quad \Gamma \overset{d_2}{\vdash} B : \mathbf{Type} \qquad \Gamma \overset{d_3}{\vdash} c : (El(A))El(B) \quad (A, B, c) \in C}{\Gamma \vdash A <_c B : \mathbf{Type}}\,(ST.1)$

  on a $\Psi(d) \equiv$

$$\dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} A_1 : \mathbf{Type} \qquad \dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} B_2 : \mathbf{Type} \qquad \overset{?_1}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash B_2 : \mathbf{Type}}\,(3.3) \qquad \dfrac{\dfrac{\Gamma_3 \overset{\Psi(d_3)}{\vdash} c_3 : (El(A_3))El(B_3) \qquad \overset{?_2}{\vdash} \Gamma_3 = \Gamma_1}{\Gamma_1 \vdash c_3 : (El(A_3))El(B_3)}\,(3.3) \qquad \Gamma_1 \overset{?_3}{\vdash} (El(A_3))El(B_3) = (El(A_1))El(B_2)}{\Gamma_1 \vdash c_3 : (El(A_1))El(B_2)}\,(3.1)}{\Gamma_1 \vdash A_1 <_{c_3} B_2 : \mathbf{Type}}\,(ST.1)$$

Les égalités dont on a besoin sont contenues dans
$\Psi_{d_1}(\Gamma \vdash \textbf{valid}) = \Psi_{d_2}(\Gamma \vdash \textbf{valid})$, $\Psi_{d_2}(\Gamma \vdash B\textbf{Type}) = \Psi_{d_3}(\Gamma \vdash B\textbf{Type})$, $\Psi_{d_1}(\Gamma \vdash A\textbf{Type}) = \Psi_{d_3}(A\textbf{Type})$.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} A <_c B : \textbf{Type} \qquad \Gamma \overset{d_2}{\vdash} A = A' : \textbf{Type}}{\Gamma \vdash A' <_c B : \textbf{Type}}$ (ST.2)

  Ce cas est représentatif pour les cas ST.3, ST.4.

$$\Psi(d) \equiv \dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} A_1 <_{c_1} B_1 : \textbf{Type} \qquad \dfrac{\Gamma_1 \overset{?_2}{\vdash} A_1 = A_2 : \textbf{Type} \qquad \dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} A_2 = A_2' : \textbf{Type} \qquad \overset{?_1}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash A_2 = A_2' : \textbf{Type}}\,(3.3)}{\Gamma_1 \vdash A_1 = A_2' : \textbf{Type}}\,(2.6)}{\Gamma_1 \vdash A_2' <_{c_1} B_1 : \textbf{Type}}\,(\text{ST.2})$$

Les égalités dont on a besoin sont contenues dans
$\Psi_{d_1}(\Gamma \vdash A\textbf{Type}) = \Psi_{d_2}(\Gamma \vdash A\textbf{Type})$.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} A <_c B : \textbf{Type} \qquad \Gamma \overset{d_2}{\vdash} c = c' : (El(A))El(B)}{\Gamma \vdash A <_{c'} B : \textbf{Type}}$ (ST.5)

  on a $\Psi(d) \equiv$

$$\dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} A_1 <_{c_1} B_1 : \textbf{Type} \qquad \dfrac{\Gamma_1 \overset{?_1}{\vdash} c_1 = c_2 : (El(A_1))El(B_1) \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} c_2 = c_2' : (El(A_2))El(B_2) \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash c_2 = c_2' : (El(A_2))El(B_2)}\,(3.3) \qquad \Gamma_1 \overset{?_3}{\vdash} (El(A_2))El(B_2) = (El(A_1))El(B_1)}{\Gamma_1 \vdash c_2 = c_2' : (El(A_1))El(B_1)}\,(3.2)}{\Gamma_1 \vdash c_1 = c_2' : (El(A_1))El(B_1)}\,(2.6)}{\Gamma_1 \vdash A_1 <_{c_2'} B_1 : \textbf{Type}}\,(\text{ST.5})$$

Les égalités dont on a besoin sont contenues dans
$\Psi_{d_1}(\Gamma \vdash c : (El(A))El(B)) = \Psi_{d_2}(\Gamma \vdash c : (El(A))El(B))$.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} Q' <_{c'} Q \qquad \Gamma, x' : Q' \overset{d_2}{\vdash} [c'(x')/x]K <_{c''} K' \qquad \Gamma, x : Q \overset{d_3}{\vdash} K\textbf{kind}}{\Gamma \vdash (x : Q)K <_c (x' : Q')K'}$ (SK.4)

avec $c \equiv [f : (x : Q)K][x' : Q']c''(f(c'(x')))$.
Ce cas est représentatif des cas SK.2, SK.3.

Considérons la dérivation $D_2 \equiv$

$$\dfrac{\dfrac{\Gamma_2, x' : Q'_2 \overset{\Psi(d_2)}{\vdash} [c'_2(x')/x]K_2 <_{c''_2} K'_2 \qquad \overset{?_1}{\vdash} \Gamma_2, x' : Q'_2 = \Gamma_1, x' : Q'_1}{\Gamma_1, x' : Q'_1 \vdash [c'_2(x')/x]K_2 <_{c''_2} K'_2}\,(3.3) \qquad \Gamma_1, x' : Q'_1 \overset{?_2}{\vdash} [c'_2(x')/x]K_2 = [c'_1(x')/x]K_3}{\Gamma_1, x' : Q'_1 \vdash [c'_1(x')/x]K_3 <_{c''_2} K'_2}\,\text{(SK.6)}$$

ainsi que $D_3 \equiv$

$$\dfrac{\Gamma_3, x : Q_3 \overset{\Psi(d_3)}{\vdash} K_3\textbf{kind} \qquad \overset{?_3}{\vdash} \Gamma_3, x : Q_3 = \Gamma_1, x : Q_1}{\Gamma_1, x : Q_1 \vdash K_3\textbf{kind}}\,(3.3)$$

Finalement on a $\Psi(d) \equiv$

$$\dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} Q'_1 <_{c'_1} Q_1 \qquad \Gamma_1, x' : Q'_1 \overset{D_2}{\vdash} [c'_1(x')/x]K_3 <_{c''_2} K'_2 \qquad \Gamma_1, x : Q_1 \overset{D_3}{\vdash} K_3\textbf{kind}}{\Gamma_1 \vdash (x : Q_1)K_3 <_{c^*} (x' : Q'_1)K'_2}\,\text{(SK.4)}$$

avec $c^* \equiv [f : (x : Q_1)K_3][x' : Q'_1]c''_2(f(c'_1(x')))$.

Les égalités $?_1$ et $?_3$ sont contenues dans
$\Psi_{d_1}(\Gamma \vdash Q'\textbf{kind}) = \Psi_{d_2}(\Gamma \vdash Q'\textbf{kind})$, $\Psi_{d_1}(\Gamma, x : Q \vdash \textbf{valid}) = \Psi_{d_3}(\Gamma, x : Q \vdash \textbf{valid})$.
L'égalité $?_2$ demande davantage d'effort. Rien ne nous affirme que $\Psi_{d_2}([c'(x')/x]K)$ a la même structure que les substitutions.

- Les cas SK.5-SK.7 sont similaires aux cas ST.2-ST.4.

- Cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} f : (x : Q')K \qquad \Gamma \overset{d_2}{\vdash} k : Q \qquad \Gamma \overset{d_3}{\vdash} Q <_c Q'}{\Gamma \vdash f(k) : [c(k)/x]K}\,\text{(CA.1)}$

on a $\Psi(d) \equiv$

$$\dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} f_1 : (x : Q'_1)K_1 \qquad \dfrac{\dfrac{\dfrac{\Gamma_3 \overset{\mathbf{co}(\Psi(d_3))}{\vdash} c_3 : (Q_3)Q'_3 \qquad \overset{?_1}{\vdash} \Gamma_3 = \Gamma_1}{\Gamma_1 \vdash c_3 : (Q_3)Q'_3}\,(3.3) \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2 : Q_2 \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2 : Q_2}\,(3.3) \qquad \Gamma_1 \overset{?_3}{\vdash} Q_2 = Q_3}{\Gamma_1 \vdash k_2 : Q_3}\,(3.1)}{\Gamma_1 \vdash c_3(k_2) : Q'_3}\,(5.5) \qquad \Gamma_1 \overset{?_4}{\vdash} Q'_3 = Q'_1}{\Gamma_1 \vdash c_3(k_2) : Q'_1}\,(3.1)}{\Gamma_1 \vdash f_1(c_3(k_2)) : [c_3(k_2)/x]K_1}\,(5.5)$$

Les égalités dont on a besoin sont contenues dans
$\Psi_{d_1}(\Gamma \vdash Q'\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash Q'\mathbf{kind})$, $\Psi_{d_1}(\Gamma \vdash Q\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash Q\mathbf{kind})$, $\Psi_{d_2}(\Gamma \vdash Q\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash Q\mathbf{kind})$.

- Le cas CA.2 est pratiquement similaire au cas CA.1.

- Le cas CD est similaire au cas CA.1 auquel on rajoute à la fin de la transformation une application de la règle 2.4.

- Cas $d \equiv$

$$\dfrac{\begin{array}{c} PSCH_X(\bar{\Theta}) \qquad \Gamma, X : \mathbf{Type} \overset{d_{1_i}}{\vdash} \Theta_i\mathbf{kind} \qquad \Gamma \overset{d_2}{\vdash} C : (M^X[\bar{\Theta}])\mathbf{Type} \\ \Gamma \overset{d_{3_i}}{\vdash} f_i : \Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]] \qquad \Gamma \overset{d_{4_j}}{\vdash} a_j : M_j \qquad 1 \leq j \leq m \quad 1 \leq i \leq n \end{array}}{\Gamma \vdash E^X[\bar{\Theta}](C, \bar{f}, \iota_i^X[\bar{\Theta}](\bar{a})) = f_i(\bar{a}, \Phi_1^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_1], \ldots, \Phi_k^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_k]) : C(\iota_i^X[\bar{\Theta}](\bar{a}))}\,(\text{E-}\textsc{eq})$$

Avec $\bar{\Theta}$ de la forme $< \Theta_1, \ldots, \Theta_1 >$, $\bar{f}$ qui représente $f_1, \ldots, f_n$, $\bar{a}$ qui représente $a_1, \ldots, a_m$ et $ARITY_X(\Theta_i) \equiv \Phi_1, \ldots, \Phi_k$. Rappelons que Θ_i est de la forme $(x_1 : M_1) \ldots (x_m : M_m)X$.

Notons que $\Psi_{d_{1_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind}) \equiv \Gamma_{1_i}, X_{1_i} : \mathbf{Type} \vdash \Theta_?\mathbf{kind}$.
Pour $v = 1$ nous prenons $D_{1_1} \equiv \Psi(d_{1_1})$ et pour $2 \leq v \leq n$, considérons les dérivations $D_{1_v} \equiv$

$$\dfrac{\Gamma_{1v}, X_{1v} : \mathbf{Type} \overset{\Psi(d_{1v})}{\vdash} \Theta_{v1v}\mathbf{kind} \qquad \overset{?}{\vdash} \Gamma_{1v}, X_{1v} : \mathbf{Type} = \Gamma_{11}, X_{11} : \mathbf{Type}}{\Gamma_{11}, X_{11} : \mathbf{Type} \vdash \Theta_{v1v}\mathbf{kind}}\,(3.3)$$

Notons que
$\Psi_{d_2}(\Gamma \vdash C : (M^X[< \Theta_1, \ldots, \Theta_1 >])\textbf{Type}) \equiv$
$\Gamma_2 \vdash C_2 : (M^X[< \Theta_{1_2}, \ldots, \Theta_{n_2} >])\textbf{Type}$.

Considérons la dérivation $h_1 \equiv$

$$\dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} C_2 : (M^X[< \Theta_{1_2}, \ldots, \Theta_{n_2} >])\textbf{Type} \qquad \overset{?}{\vdash} \Gamma_2 = \Gamma_{1_1}}{\Gamma_{1_1} \vdash C_2 : (M^X[< \Theta_{1_2}, \ldots, \Theta_{n_2} >])\textbf{Type}}\,(3.3) \qquad \Gamma_{1_1} \overset{?}{\vdash} (M^X[< \Theta_{1_2}, \ldots, \Theta_{n_2} >])\textbf{Type} = (M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_2} >])\textbf{Type}}{\Gamma_{1_1} \vdash C_2 : (M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_2} >])\textbf{Type}}\,(3.1)$$

Puis la dérivation $D_2 \equiv$

$$\dfrac{\Gamma_{1_1} \overset{h_1}{\vdash} C_2 : (M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_2} >])\textbf{Type} \qquad \ldots \Gamma_{1_1} \overset{?}{\vdash} (M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_2} >])\textbf{Type} = (M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >])\textbf{Type}}{\Gamma_{1_1} \vdash C_2 : (M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >])\textbf{Type}}\,(3.1)$$

Notons que
$\Psi_{d_{3_i}}(\Gamma \vdash f_i : \Theta_i^\circ[M^X[< \Theta_1, \ldots, \Theta_n >], C, \iota_i^X[< \Theta_1, \ldots, \Theta_n >]]) \equiv$
$\Gamma_{3_i} \vdash f_{i_{3_i}} : \Theta_i^\circ[M^X[< \Theta_{1_{3_i}}, \ldots, \Theta_{n_{3_i}} >], C, \iota_i^X[< \Theta_{1_{3_i}}, \ldots, \Theta_{n_{3_i}} >]]$.

Pour $1 \leq v \leq n$ considérons les dérivations $D_{3_v} \equiv$

$$\dfrac{\dfrac{\Gamma_{3_v} \overset{\Psi(d_{3_v})}{\vdash} f_{v_{3_v}} : \Theta_v^\circ[M^X[< \Theta_{1_{3_v}}, \ldots, \Theta_{n_{3_v}} >], C, \iota_v^X[< \Theta_{1_{3_v}}, \ldots, \Theta_{n_{3_v}} >]] \qquad \overset{?}{\vdash} \Gamma_{3_v} = \Gamma_{1_1}}{\Gamma_{1_1} \vdash f_{v_{3_v}} : \Theta_v^\circ[M^X[< \Theta_{1_{3_v}}, \ldots, \Theta_{n_{3_v}} >], C, \iota_v^X[< \Theta_{1_{3_v}}, \ldots, \Theta_{n_{3_v}} >]]}\,(3.3) \qquad J}{\Gamma_{1_1} \vdash f_{v_{3_v}} : \Theta_v^\circ[M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >], C_2, \iota_v^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >]]}\,(3.1)$$

avec
$J \equiv \Gamma_{1_1} \overset{?}{\vdash} \Theta_v^\circ[M^X[< \Theta_{1_{3_v}}, \ldots, \Theta_{n_{3_v}} >], C, \iota_v^X[< \Theta_{1_{3_v}}, \ldots, \Theta_{n_{3_v}} >]] =$
$\Theta_v^\circ[M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >], C_2, \iota_v^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >]]$

Notons que $\Psi_{d_{4_j}}(\Gamma \vdash a_j : M_j) \equiv \Gamma_{4_j} \vdash a_{j_{4_j}} : M_{j_{4_j}}$.
Pour $1 \leq p \leq m$, considérons les dérivations $D_{4p} \equiv$

$$\dfrac{\dfrac{\Gamma_{4_p} \overset{\Psi(d_{4p})}{\vdash} a_{p_{4_p}} : M_{p_{4_p}} \qquad \overset{?}{\vdash} \Gamma_{4_p} = \Gamma_{1_1}}{\Gamma_{1_1} \vdash a_{p_{4_p}} : M_{p_{4_p}}}\,(3.3) \qquad \Gamma_{1_1} \overset{?}{\vdash} M_{p_{4_p}} = M_{p_{1_p}}}{\Gamma_{1_1} \vdash a_{p_{4_p}} : M_{p_{1_p}}}\,(3.1)$$

Finalement on a $\Psi(d) \equiv$

$$\dfrac{\begin{array}{c} PSCH_X(\bar{\Theta}') \\ \Gamma_{1_1}, X_{1_1} : \textbf{Type} \overset{D_{1_v}}{\vdash} \Theta_{v_{1_v}} \textbf{kind} \qquad \Gamma_{1_1} \overset{D_2}{\vdash} C_2 : (M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >])\textbf{Type} \\ \Gamma_{1_1} \overset{D_{3_v}}{\vdash} f_{v_{3_v}} : \Theta_v^{\circ}[M^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >], C_2, \iota_v^X[< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >]] \\ \Gamma_{1_1} \overset{D_{4_p}}{\vdash} a_{p_{4_p}} : M_{p_{1_p}} \qquad 1 \leq p \leq m \quad 1 \leq v \leq n \end{array}}{\Gamma \vdash E^X[\bar{\Theta}'](C_2, \bar{f}', \iota_v^X[\bar{\Theta}'](\bar{a}')) = f_{v_{3_v}}(\bar{a}', \Phi_1^{\natural}[M^X[\bar{\Theta}'], C_2, E^X[\bar{\Theta}'](C_2, \bar{f}'), a_{1_{4_1}}], \ldots, \Phi_k^{\natural}[M^X[\bar{\Theta}'], C_2, E^X[\bar{\Theta}'](C_2, \bar{f}'), a_{k_{4_k}}]) : C_2(\iota_i^X[\bar{\Theta}'](\bar{a}'))} \text{ (E-EQ)}$$

Avec $\bar{\Theta}'$ de la forme $< \Theta_{1_{1_1}}, \ldots, \Theta_{n_{1_n}} >$, $\bar{f}'$ qui représente $f_{1_{3_1}}, \ldots, f_{n_{3_n}}$, $\bar{a}'$ qui représente $a_{1_{4_1}}, \ldots, a_{m_{4_m}}$.

Les égalités dont on a besoin sont prouvables à partir de $\Psi_{d_{1_i}}(\Gamma, X : \textbf{Type} \vdash \Theta_i\textbf{kind}) = \Psi_{d_2}(\Gamma, X : \textbf{Type} \vdash \Theta_i\textbf{kind})$, $\Psi_{d_2}(\Gamma \vdash C : (M^X[< \Theta_1, \ldots, \Theta_1 >])\textbf{Type}) = \Psi_{d_{3_i}}(\Gamma \vdash C : (M^X[< \Theta_1, \ldots, \Theta_1 >])\textbf{Type})$, $\Psi_{d_{1_i}}(\Gamma \vdash M_j\textbf{kind}) = \Psi_{d_{4_j}}(\Gamma \vdash M_j\textbf{kind})$.

- Cas $d \equiv$

$$\dfrac{\Gamma \overset{d_1}{\vdash} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma \overset{d_2}{\vdash} E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m]) : (F_n)F_m \qquad \Gamma \overset{d_3}{\vdash} k : F_n}{\Gamma \vdash E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m])(k)) = E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A} \text{ (ϑ-eq)}$$

Pour plus de clarté, nous désignons ici par $\Psi_h(c)$, l'occurrence du constituant c provenant de la dérivation $\Psi(h)$, pour une dérivation h donnée.

Considérons la dérivation $D_1 \equiv$

$$\dfrac{\dfrac{\Psi_{d_2}(\Gamma) \overset{\Psi(d_2)}{\vdash} \Psi_{d_2}(E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m]) : (F_n)F_m) \qquad \overset{?_1}{\vdash} \Psi_{d_2}(\Gamma) = \Psi_{d_1}(\Gamma)}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_2}(E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m]) : (F_n)F_m)} \text{ (3.3)} \qquad \Psi_{d_1}(\Gamma) \overset{?_2}{\vdash} (\Psi_{d_2}(F_n))\Psi_{d_2}(F_m) = (\Psi_{d_2}(F_n))\Psi_{d_1}(F_m)}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_2}(E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m])) : (\Psi_{d_2}(F_n))\Psi_{d_1}(F_m)} \text{ (3.1)}$$

ainsi que $D_2 \equiv$

$$\cfrac{\cfrac{\Psi_{d_3}(\Gamma) \overset{\Psi(d_3)}{\vdash} \Psi_{d_3}(k : F_n) \qquad \overset{?_3}{\vdash} \Psi_{d_3}(\Gamma) = \Psi_{d_1}(\Gamma)}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_3}(k : F_n)}\,(3.3) \qquad \Psi_{d_1}(\Gamma) \overset{?_4}{\vdash} \Psi_{d_3}(F_n) = \Psi_{d_2}(F_n)}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_3}(k) : \Psi_{d_2}(F_n)}\,(3.1)$$

on a $\Psi(d) \equiv$

$$\cfrac{\begin{array}{c}\Psi_{d_1}(\Gamma) \overset{\Psi(d_1)}{\vdash} \Psi_{d_1}(\Gamma \overset{d_1}{\vdash} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A) \\ \Psi_{d_1}(\Gamma) \overset{D_1}{\vdash} \Psi_{d_2}(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])) : (\Psi_{d_2}(F_n))\Psi_{d_1}(F_m) \\ \Psi_{d_1}(\Gamma) \overset{D_2}{\vdash} \Psi_{d_3}(k) : \Psi_{d_2}(F_n)\end{array}}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(E[\bar{\Theta}_m](C, a_1, ..., a_m))(\Psi_{d_2}(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]))(\Psi_{d_3}(k))) = E[\Psi_{d_2}(\bar{\Theta}_n)](C_2', \Psi_{d_1}(a_{\sigma(1)}), \ldots, \Psi_{d_1}(a_{\sigma(n)}))(\Psi_{d_3}(k)) : \Psi_{d_1}(A)}\,(\vartheta\text{-EQ})$$

où $C_2 \equiv [x : \Psi_{d_2}(F_n)]\Psi_{d_1}(A) : (x : \Psi_{d_2}(F_n))$**Type**.

Les égalités dont on a besoin sont prouvables à partir de $\Psi_{d_1}(\Gamma \vdash F_m\textbf{kind}) = \Psi_{d_2}(\Gamma \vdash F_m\textbf{kind})$, $\Psi_{d_1}(\Gamma \vdash \textbf{valid}) = \Psi_{d_3}(\Gamma \vdash \textbf{valid})$, $\Psi_{d_2}(\Gamma \vdash F_n\textbf{kind}) = \Psi_{d_3}(\Gamma \vdash F_n\textbf{kind})$.

Il faut tout de même remarquer qu'il n'y a aucun doute par rapport au fait que $\Psi_{d_1}(\bar{\Theta}_m) = \Psi_{d_2}(\bar{\Theta}_m)$, car rappelons-le, $\bar{\Theta}_m \equiv < \underbrace{X, \ldots, X}_{m fois} >$. X étant une variable, elle demeure inchangée après application de Ψ.

4.2.4.2 Ψ et les dérivations

Nous explorons dans cette section certaines propriétés relatives à la transformation Ψ et plus particulièrement, le lien qui existe entre Ψ et les algorithmes que nous avons introduits dans la section 4.2.3.3. Pour cela, nous considérons ces algorithmes dans le même ordre qu'ils apparaissent dans la section 4.2.3.3.

Lemme 4.40. *Soit une dérivation* d *telle que* $\Psi(d)$ *soit définie.*

(a) $\Psi(d)$ *se termine par la même règle que* d.

(b) Si d_0 est une sous-dérivation de d, alors $\Psi(d_0)$ est définie.

(c) Soit d_0 une sous-dérivation de d du jugement J et une dérivation d'_0 telle que $d_0 \sim d'_0$. La dérivation d' est obtenue à partir de d en remplaçant d_0 par d'_0. Si $\Psi(d'_0)$ est définie et $\Psi_{d_0}(J) = \Psi_{d'_0}(J)$ alors $\Psi(d')$ est définie et $\Psi(d) \sim \Psi(d')$.

Démonstration.

(a) Vrai par definition de Ψ.

(b) Vrai par definition de Ψ.

(c) Preuve par induction sur la taille de d.

Cas de base. Quand $d \equiv d_0$, évident.

Pas d'induction. Considérons le cas :

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} A <_c B : \mathbf{Type} \qquad \Gamma \overset{d_2}{\vdash} A = A' : \mathbf{Type}}{\Gamma \vdash A' <_c B : \mathbf{Type}} \text{(ST.2)}$$

Soit la dérivation d'_1 telle que $d_1 \sim d'_1$. On considère la dérivation d' dans laquelle d_1 est remplacée par d'_1. Par hypothèse $\Psi(d'_1)$ est définie. Par définition de Ψ et sachant par hypothèse que $\Psi_{d_1}(\Gamma \vdash A <_c B : \mathbf{Type}) = \Psi_{d'_1}(\Gamma \vdash A <_c B : \mathbf{Type})$, on a $\Psi_{d'} \equiv$

$$\frac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} A_1 <_{c_1} B_1 : \mathbf{Type} \qquad \dfrac{\Gamma_1 \overset{?_2}{\vdash} A_1 = A_2 : \mathbf{Type} \qquad \dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} A_2 = A'_2 : \mathbf{Type} \qquad \overset{?_1}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash A_2 = A'_2 : \mathbf{Type}} (3.3)}{\Gamma_1 \vdash A_1 = A'_2 : \mathbf{Type}} (2.6)}{\Gamma_1 \vdash A'_2 <_{c_1} B_1 : \mathbf{Type}} \text{(ST.2)}$$

d'où $\Psi(d) \sim \Psi(d')$.

□

Lemme 4.41. *Soit d une dérivation d'un des jugements $\Gamma, \Gamma' \vdash J$; $\Gamma, x : K, \Gamma' \vdash J$ ou $\Gamma \vdash (x : K_1)K_2$ dans $UTT^r[R]^-$. Si $\Psi(d)$ est définie alors respectivement :*

(a) $\Psi(V^{\circ}_{\Gamma_1}(d))$ *est définie et* $\Psi_{V^{\circ}_{\Gamma_1}(d)}(\Gamma \vdash \textbf{\textit{valid}}) = \Psi_d(\Gamma \vdash \textbf{\textit{valid}})$ *;*

(b) $\Psi(K^{\circ}_{\Gamma_1}(d))$ *est définie et* $\Psi_{K^{\circ}_{\Gamma_1}(d)}(\Gamma \vdash K\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K\textbf{\textit{kind}})$ *;*

(c) $\Psi(p\_^{\circ}(d))$ *est définie et* $\Psi_{p\_^{\circ}(d)}(\Gamma, x : K_1 \vdash K_2\textbf{\textit{kind}}) = \Psi_d(\Gamma, x : K_1 \vdash K_2\textbf{\textit{kind}})$*.*

Démonstration. Les dérivations $V^{\circ}_{\Gamma_1}(d)$, $K^{\circ}_{\Gamma_1}(d)$ et $p\_^{\circ}(d)$ sont des sous-dérivations de d, par conséquent $\Psi(V^{\circ}_{\Gamma_1}(d))$, $\Psi(K^{\circ}_{\Gamma_1}(d))$ et $\Psi(p\_^{\circ}(d))$ sont définies (lemme 4.40). On prouve l'égalité des jugements par induction structurelle.

Cas de base. Cas de la règle 1.1, évident.

Pas d'induction. Considérons le cas :

$$d \equiv \frac{\Gamma, \Gamma' \overset{d_1}{\vdash} A <_c B : \textbf{Type} \qquad \Gamma, \Gamma' \overset{d_2}{\vdash} A = A' : \textbf{Type}}{\Gamma, \Gamma' \vdash A' <_c B : \textbf{Type}} \text{(ST.2)}$$

Par définition on sait que $\Psi(d) \equiv$

$$\frac{\Gamma_1, \Gamma_1' \overset{\Psi(d_1)}{\vdash} A_1 <_{c_1} B_1 : \textbf{Type} \qquad \dfrac{\Gamma_1, \Gamma_1' \overset{?_2}{\vdash} A_1 = A_2 : \textbf{Type} \qquad \dfrac{\Gamma_2, \Gamma_2' \overset{\Psi(d_2)}{\vdash} A_2 = A_2' : \textbf{Type} \qquad \overset{?_1}{\vdash} \Gamma_2, \Gamma_2' = \Gamma_1, \Gamma_1'}{\Gamma_1, \Gamma_1' \vdash A_2 = A_2' : \textbf{Type}} \text{(3.3)}}{\Gamma_1, \Gamma_1' \vdash A_1 = A_2' : \textbf{Type}} \text{(2.6)}}{\Gamma_1, \Gamma_1' \vdash A_2' <_{c_1} B_1 : \textbf{Type}} \text{(ST.2)}$$

On a $\Psi_d(\Gamma \vdash \textbf{valid}) = \Psi_{d_1}(\Gamma \vdash \textbf{valid})$ (conséquence de la remarque 4.39), par hypothèse d'induction $\Psi_{d_1}(\Gamma \vdash \textbf{valid}) = \Psi_{V^{\circ}_{\Gamma_1}(d_1)}(\Gamma \vdash \textbf{valid})$, or $V^{\circ}_{\Gamma_1}(d) \equiv V^{\circ}_{\Gamma_1}(d_1)$, alors $\Psi_{V^{\circ}_{\Gamma_1}(d_1)}(\Gamma \vdash \textbf{valid}) = \Psi_{V^{\circ}_{\Gamma_1}(d)}(\Gamma \vdash \textbf{valid})$. On peut conclure par $\Psi_d(\Gamma \vdash \textbf{valid}) = \Psi_{V^{\circ}_{\Gamma_1}(d)}(\Gamma \vdash \textbf{valid})$.

□

Lemme 4.42. *Soit d une $UTT^r[R]_w$-dérivation et $\boldsymbol{E}^{\circ}$ l'algorithme d'élimination vu au lemme 4.19. Si $\Psi(d)$ est définie, alors $\Psi(\boldsymbol{E}^{\circ}(d))$ est également définie et $\Psi(\boldsymbol{E}^{\circ}(d)) \sim \Psi(d)$.*

Démonstration. La preuve se fait par induction sur la taille de la dérivation de la prémisse gauche. Nous devons considérer tous les sous cas de l'algorithme $\mathbf{E}^{\circ}$, ici nous prendrons comme illustration les cas de l'élimination de wkn ainsi que l'élimination des substitutions simples.

- **Elimination de** wkn.
  Prenons par exemple le cas où d_1 se termine par la règle 5.7.

$$d \equiv \quad \frac{\Gamma, \Delta \overset{d_1}{\vdash} ([x:K]k')k = [k/x]k' : [k/x]K' \qquad \Gamma, \Gamma'' \overset{d_2}{\vdash} \textbf{valid}}{\Gamma, \Gamma'', \Delta, x:K \vdash ([x:K]k')k = [k/x]k' : [k/x]K'} \text{(wkn)}$$

avec $d_1 \equiv \dfrac{\Gamma, \Delta, x:K \overset{e_1}{\vdash} k' : K' \qquad \Gamma, \Delta \overset{e_2}{\vdash} k : K}{\Gamma, \Delta \vdash ([x:K]k')k = [k/x]k' : [k/x]K'}$ (5.7)

et d_1 et d_2 ne contiennent pas wkn.

Soit les dérivations

$$e_1' \equiv \quad \frac{\Gamma, \Delta, x:K \overset{e_1}{\vdash} k' : K' \qquad \Gamma, \Gamma'' \overset{d_2}{\vdash} \textbf{valid}}{\Gamma, \Gamma'', \Delta, x:K \vdash k' : K'} \text{(wkn)} \text{ et}$$

$$e_2' \equiv \quad \frac{\Gamma, \Delta \overset{e_2}{\vdash} k : K \qquad \Gamma, \Gamma'' \overset{d_2}{\vdash} \textbf{valid}}{\Gamma, \Gamma'', \Delta \vdash k : K} \text{(wkn)}$$

Les dérivations $\Psi(e_1')$ et $\Psi(e_2')$ ont les formes suivantes :

$$\Psi(e_1') \equiv \frac{\Psi_{e_1}(\Gamma)\Psi_{e_1}(\Delta), x:\Psi_{e_1}(K) \overset{\Psi(e_1)}{\vdash} \Psi_{e_1}(k') : \Psi_{e_1}(K') \qquad \dfrac{\Psi_{d_2}(\Gamma), \Psi_{d_2}(\Gamma'') \overset{\Psi(d_2)}{\vdash} \textbf{valid} \qquad \overset{E_1}{\vdash} \Psi_{d_2}(\Gamma) = \Psi_{e_1}(\Gamma)}{\Psi_{e_1}(\Gamma), \Psi_{d_2}(\Gamma'') \vdash \textbf{valid}} \text{(3.3)}}{\Psi_{e_1}(\Gamma), \Psi_{d_2}(\Gamma''), \Psi_{e_1}(\Delta), x:\Psi_{e_1}(K) \vdash \Psi_{e_1}(k') : \Psi_{e_1}(K')} \text{(wkn)}$$

$$\Psi(e_2') \equiv \frac{\Psi_{e_2}(\Gamma)\Psi_{e_2}(\Delta) \overset{\Psi(e_2)}{\vdash} \Psi_{e_2}(k) : \Psi_{e_2}(K) \qquad \dfrac{\Psi_{d_2}(\Gamma), \Psi_{d_2}(\Gamma'') \overset{\Psi(d_2)}{\vdash} \textbf{valid} \qquad \overset{E_2}{\vdash} \Psi_{d_2}(\Gamma) = \Psi_{e_2}(\Gamma)}{\Psi_{e_2}(\Gamma), \Psi_{d_2}(\Gamma'') \vdash \textbf{valid}} \text{(3.3)}}{\Psi_{e_2}(\Gamma), \Psi_{d_2}(\Gamma''), \Psi_{e_2}(\Delta) \vdash \Psi_{e_2}(k) : \Psi_{e_2}(K)} \text{(wkn)}$$

Par hypothèse $\Psi(d)$ est définie, alors $\Psi(d_1)$ et $\Psi(d_2)$ le sont également. Par rapport à la définition de $\Psi(d)$, on peut déduire les égalités suivantes $\Psi_{d_2}(\Gamma) = \Psi_{e_1}(\Gamma) = \Psi_{e_2}(\Gamma)$ qui contiennent les égalités dénotées par E_1 et E_2. D'où $\Psi(e_1')$ et $\Psi(e_2')$ sont définies.

On sait que $\mathbf{E}^\circ(d)$ est de la forme suivante :

$$\mathbf{E}^{\circ}(d) \equiv \quad \frac{\Gamma, \Gamma'', \Delta, x : K \overset{\mathbf{E}^{\circ}(e'_1)}{\vdash} k' : K' \qquad \Gamma, \Gamma'', \Delta \overset{\mathbf{E}^{\circ}(e'_2)}{\vdash} k : K}{\Gamma, \Gamma'', \Delta \vdash ([x : K]k')k = [k/x]k' : [k/x]K'} \; (5.7)$$

Condidérons maintenant les dérivations suivantes et définissons $\Psi(\mathbf{E}^{\circ}(d))$:

$$g_1 \equiv \frac{\Psi_{e'_2}(\Gamma), \Psi_{e'_2}(\Gamma''), \Psi_{e'_2}(\Delta) \overset{\Psi(\mathbf{E}^{\circ}(e'_2))}{\vdash} \Psi_{e'_2}(k) : \Psi_{e'_2}(K) \qquad \overset{E_3}{\vdash} \Psi_{e'_2}(\Gamma), \Psi_{e'_2}(\Gamma''), \Psi_{e'_2}(\Delta) = \Psi_{e'_1}(\Gamma), \Psi_{e'_1}(\Gamma''), \Psi_{e'_1}(\Delta)}{\Psi_{e'_1}(\Gamma), \Psi_{e'_1}(\Gamma''), \Psi_{e'_1}(\Delta) \vdash \Psi_{e'_2}(k) : \Psi_{e'_2}(K)} \; (3.3)$$

$$g_2 \equiv \frac{\Psi_{e'_1}(\Gamma), \Psi_{e'_1}(\Gamma''), \Psi_{e'_1}(\Delta) \overset{g_1}{\vdash} \Psi_{e'_2}(k) : \Psi_{e'_2}(K) \qquad \Psi_{e'_1}(\Gamma), \Psi_{e'_1}(\Gamma''), \Psi_{e'_1}(\Delta) \overset{E_4}{\vdash} \Psi_{e'_2}(K) = \Psi_{e'_1}(K)}{\Psi_{e'_1}(\Gamma), \Psi_{e'_1}(\Gamma''), \Psi_{e'_1}(\Delta) \vdash \Psi_{e'_2}(k) : \Psi_{e'_1}(K)} \; (3.1)$$

$$\Psi(\mathbf{E}^{\circ}(d)) \equiv \frac{\Psi_{e'_1}(\Gamma), \Psi_{e'_1}(\Gamma''), \Psi_{e'_1}(\Delta), x : \Psi_{e'_1}(K) \overset{\Psi(\mathbf{E}^{\circ}(e'_1))}{\vdash} \Psi_{e'_1}(k') : \Psi_{e'_1}(K)' \qquad \Psi_{e'_1}(\Gamma), \Psi_{e'_1}(\Gamma''), \Psi_{e'_1}(\Delta) \overset{g_2}{\vdash} \Psi_{e'_2}(k) : \Psi_{e'_1}(K)}{\Psi_{e'_1}(\Gamma), \Psi_{e'_1}(\Gamma''), \Psi_{e'_1}(\Delta) \vdash ([x : \Psi_{e'_1}(K)]\Psi_{e'_1}(k'))\Psi_{e'_2}(k) = [\Psi_{e'_2}(k)/x]\Psi_{e'_1}(k') : [\Psi_{e'_2}(k)/x]\Psi_{e'_1}(K)'} \; (5.7)$$

- Commençons par montrer que $\Psi(\mathbf{E}^{\circ}(d))$ est définie. Pour cela nous devons prouver que les égalités E_3 et E_4 sont dérivables. Il nous suffit de prouver que $\Psi_{e'_2}(\Gamma, \Gamma'', \Delta \vdash K\mathbf{kind}) = \Psi_{e'_1}(\Gamma, \Gamma'', \Delta \vdash K\mathbf{kind})$.
  En appliquant l'hypothèse d'induction à e'_1 et e'_2, à savoir $\Psi(e'_1) \sim \Psi(\mathbf{E}^{\circ}(e'_1))$ et $\Psi(e'_2) \sim \Psi(\mathbf{E}^{\circ}(e'_2))$, et par définition de $\Psi(e'_1)$, $\Psi(e'_2)$ et $\Psi(d)$, on peut prouver les égalités :
  $\Psi_{e'_2}(\Gamma, \Gamma'', \Delta \vdash K\mathbf{kind}) = \Psi_{e_1}(\Gamma), \Psi_{d_2}(\Gamma''), \Psi_{e_1}(\Delta) \vdash \Psi_{e_1}(K)\mathbf{kind})$,
  et $\Psi_{e'_1}(\Gamma, \Gamma'', \Delta \vdash K\mathbf{kind}) = \Psi_{e_1}(\Gamma), \Psi_{d_2}(\Gamma''), \Psi_{e_1}(\Delta) \vdash \Psi_{e_1}(K)\mathbf{kind})$.
  Des deux égalités précédentes, on déduit l'égalité recherchée.
  La dérivation $\Psi(\mathbf{E}^{\circ}(d))$ est donc bien définie. Notons que l'on peut lui appliquer l'algorithme $\mathbf{E}^{\circ}$ si on veut une $UTT^r[R]^-$-dérivation.

- Montrons maintenant que
  $\Psi_d(\Gamma, \Gamma'', \Delta \vdash ([x : K]k')k = [k/x]k' : [k/x]K') = \Psi_{\mathbf{E}^{\circ}(d)}(\Gamma, \Gamma'', \Delta \vdash ([x : K]k')k = [k/x]k' : [k/x]K')$.

  Par définition de $\Psi(d)$, on peut dire que (1) $\Psi_d(\Gamma, \Gamma'', \Delta \vdash ([x : K]k')k = [k/x]k' : [k/x]K') = \Psi_{e_1}(\Gamma), \Psi_{d_2}(\Gamma''), \Psi_{e_1}(\Delta) \vdash ([x : \Psi_{e_1}(K)]\Psi_{e_1}(k'))\Psi_{e_2}(k) = [\Psi_{e_2}(k)/x]\Psi_{e_1}(k') : [\Psi_{e_2}(k)/x]\Psi_{e_1}(K')$.
  Par hypothèse d'induction on sait que $\Psi(e'_1) \sim \Psi(\mathbf{E}^{\circ}(e'_1))$ et $\Psi(e'_2) \sim$

$\Psi(\mathbf{E}^\circ(e'_2))$, de plus par définition de $\Psi(\mathbf{E}^\circ(d))$ et grâce au lemme 4.40, on peut dire que (2) $\Psi_{\mathbf{E}^\circ(d)}(\Gamma, \Gamma'', \Delta \vdash ([x:K]k')k = [k/x]k' : [k/x]K') = \Psi_{e_1}(\Gamma), \Psi_{d_2}(\Gamma''), \Psi_{e_1}(\Delta) \vdash ([x:\Psi_{e_1}(K)]\Psi_{e_1}(k'))\Psi_{e_2}(k) = [\Psi_{e_2}(k)/x]\Psi_{e_1}(k') : [\Psi_{e_2}(k)/x]\Psi_{e_1}(K')$.
Des égalités (1) et (2) on peut déduire que $\Psi_d(\Gamma, \Gamma'', \Delta \vdash ([x:K]k')k = [k/x]k' : [k/x]K') = \Psi_{\mathbf{E}^\circ(d)}(\Gamma, \Gamma'', \Delta \vdash ([x:K]k')k = [k/x]k' : [k/x]K')$. De plus les jugements $\Psi_d(\Gamma, \Gamma'', \Delta \vdash ([x:K]k')k = [k/x]k' : [k/x]K')$ et $\Psi_{\mathbf{E}^\circ(d)}(\Gamma, \Gamma'', \Delta \vdash ([x:K]k')k = [k/x]k' : [k/x]K')$ sont respectivement les conclusions des dérivations $\Psi(d)$ et $\Psi(\mathbf{E}^\circ(d))$. Par conclusion $\Psi(d) \sim \Psi(\mathbf{E}^\circ(d))$.

- **Elimination de la substitution simple** : $d \equiv \dfrac{\Gamma, x:K, \Gamma' \overset{d_1}{\vdash} J \qquad \Gamma \overset{d_2}{\vdash} k:K}{\Gamma, [k/x]\Gamma' \vdash [k/x]J}$ (sub)

  où d_1 et d_2 sont des $UTT[R]^-_w$-dérivations.

  – Prenons pour exemple le cas de la règle 6.3.
  Considérons un des cas standards (d_1 se termine par CA.1), ce cas est similaire aux autres cas à l'exception des cas où d_1 se termine par la règle 1.2 ou 1.3.

$$d \equiv \dfrac{\Gamma, x:K, \Gamma' \overset{d_1}{\vdash} f(q) : [c(q)/y]Q' \qquad \Gamma \overset{d_2}{\vdash} k:K}{\Gamma, [k/x]\Gamma' \vdash [k/x](f(q)) : [k/x]([c(q)/y]Q')} \text{(6.3)}$$

  avec

$$d_1 \equiv \dfrac{\Gamma, x:K, \Gamma' \overset{D_1}{\vdash} f:(y:Q)Q' \qquad \Gamma, x:K, \Gamma' \overset{D_2}{\vdash} q:S \qquad \Gamma, x:K, \Gamma' \overset{D_3}{\vdash} S <_c Q}{\Gamma, x:K, \Gamma' \vdash f(q) : [c(q)/y]Q'} \text{(CA.1)}$$

  Considérons la dérivation d' qui représente un pas de l'algorithme d'élimination des substitutions simples.
  Considérons la dérivation $d' \equiv$

$$\dfrac{\dfrac{\Gamma, x:K, \Gamma' \overset{D_1}{\vdash} f:(y:Q)Q' \qquad \Gamma \overset{d_2}{\vdash} k:K}{\Gamma, [k/x]\Gamma' \vdash [k/x]f : (y:[k/x]Q)[k/x]Q'} \text{(6.3)} \quad \dfrac{\Gamma, x:K, \Gamma' \overset{D_2}{\vdash} q:S \qquad \Gamma \overset{d_2}{\vdash} k:K}{\Gamma, [k/x]\Gamma' \vdash [k/x]q : [k/x]S} \text{(6.3)} \quad \dfrac{\Gamma, x:K, \Gamma' \overset{D_3}{\vdash} S <_c Q \qquad \Gamma \overset{d_2}{\vdash} k:K}{\Gamma, [k/x]\Gamma' \vdash [k/x]S <_{[k/x]c} [k/x]Q} \text{(SK.9)}}{\Gamma, [k/x]\Gamma' \vdash [k/x]f([k/x]q) : [[k/x]c([k/x]q)/y][k/x]Q'} \text{(CA.1)}$$

  Notons que $[k/x]f([k/x]q) \equiv [k/x](f(q))$ et que $[[k/x]c([k/x]q)/y][k/x]Q' \equiv [k/x]([c(q)/y]Q')$.
  Par hypothèse $\Psi(d)$ est définie, c'est donc également le cas pour $\Psi(d_1)$ (lemme 4.40) qui a par définition la structure suivante (voir section 4.2.4.1) :

$\Psi(d_1) \equiv$

$$\dfrac{\Delta_1 \overset{\Psi(D_1)}{\vdash} f_1 : (y : Q_1)Q'_1 \qquad \dfrac{\dfrac{\dfrac{\Delta_3 \overset{h}{\vdash} c_3 : (S_3)Q_3 \qquad \overset{E_1}{\vdash} \Delta_3 = \Delta_1}{\Delta_1 \vdash c_3 : (S_3)Q_3}\,(3.3) \qquad \dfrac{\dfrac{\Delta_2 \overset{\Psi(D_2)}{\vdash} q_2 : S_2 \qquad \overset{E_2}{\vdash} \Delta_2 = \Delta_1}{\Delta_1 \vdash q_2 : S_2}\,(3.3) \qquad \Delta_1 \overset{E_3}{\vdash} S_2 = S_3}{\Delta_1 \vdash q_2 : S_3}\,(3.1)}{\Delta_1 \vdash c_3(q_2) : Q_3}\,(5.5) \qquad \Delta_1 \overset{E_4}{\vdash} Q_3 = Q_1}{\Delta_1 \vdash c_3(q_2) : Q_1}\,(3.1)}{\Delta_1 \vdash f_1(c_3(q_2)) : [c_3(q_2)/y]Q'_1}\,(5.5)$$

avec $h \equiv \mathbf{co}^\circ\Psi((D_3))$ et $\Psi_{D_i}(\Gamma, x : K, \Gamma') \equiv \Gamma_i, x : K_i, \Gamma'_i \equiv \Delta_i$ pour $i = 1, 2, 3$.

Soit $\Psi_{d_2}(\Gamma \vdash k : K) \equiv \Gamma_s \vdash k_s : K_s$, on a :

$$\Psi(d) \equiv \dfrac{\Gamma_1, x : K_1, \Gamma'_1 \overset{\Psi(d_1)}{\vdash} f_1(c_3(q_2)) : [c_3(q_2)/y]Q'_1 \qquad \dfrac{\dfrac{\Gamma_s \overset{\Psi(d_2)}{\vdash} k_s : K_s \qquad \overset{E_5}{\vdash} \Gamma_s = \Gamma_1}{\Gamma_1 \vdash k_s : K_s}\,(3.3) \qquad \Gamma_1 \overset{E_6}{\vdash} K_s = K_1}{\Gamma_1 \vdash k_s : K_1}\,(3.1)}{\Gamma_1, [k_s/x]\Gamma'_1 \vdash [k_s/x]f_1([k_s/x]c_3([k_s/x]q_2)) : [[k_s/x]c_3([k_s/x]q_2)/y][k_s/x]Q'_1}\,(6.3)$$

Nous cherchons maintenant à définir $\Psi(d')$. Commençons par considérer :

$$D'_1 \equiv \quad \dfrac{\Gamma, x : K, \Gamma' \overset{D_1}{\vdash} f : (y : Q)Q' \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]f : (y : [k/x]Q)[k/x]Q'}\,(6.3)$$

on a $\Psi(D'_1) \equiv$

$$\dfrac{\Gamma_1, x : K_1, \Gamma'_1 \overset{\Psi(D_1)}{\vdash} f_1 : (y : Q_1)Q'_1 \qquad \dfrac{\dfrac{\Gamma_s \overset{\Psi(d_2)}{\vdash} k_s : K_s \qquad \overset{E_5}{\vdash} \Gamma_s = \Gamma_1}{\Gamma_1 \vdash k_s : K_s}\,(3.3) \qquad \Gamma_1 \overset{E_6}{\vdash} K_s = K_1}{\Gamma_1 \vdash k_s : K_1}\,(3.1)}{\Gamma_1, [k_s/x]\Gamma'_1 \vdash [k_s/x]f_1 : (y : [k_s/x]Q_1)[k_s/x]Q'_1}\,(6.3)$$

Les dérivations $\Psi(D_1)$ et $\Psi(d_2)$ sont définies, de plus on sait que les égalités E_5 et E_6 sont dérivables car elles apparaissent dans la dérivation $\Psi(d)$ qui est définie. $\Psi(D'_1)$ est donc bien définie.
Considérons ensuite

$$D'_2 \equiv \quad \dfrac{\Gamma, x : K, \Gamma' \overset{D_2}{\vdash} q : S \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]q : [k/x]S}\,(6.3)$$

On a $\Psi(D'_2) \equiv$

$$\dfrac{\Gamma_2, x : K_2, \Gamma_2' \overset{\Psi(D_2)}{\vdash} q_2 : S_2 \qquad \dfrac{\dfrac{\Gamma_s \overset{\Psi(d_2)}{\vdash} k_s : K_s \qquad \overset{E_7}{\vdash} \Gamma_s = \Gamma_2}{\Gamma_2 \vdash k_s : K_s}\text{(3.3)} \qquad \Gamma_2 \overset{E_8}{\vdash} K_s = K_2}{\Gamma_2 \vdash k_s : K_2}\text{(3.1)}}{\Gamma_2, [k_s/x]\Gamma_2' \vdash [k_s/x]q_2 : [k_s/x]S_2}\text{(6.3)}$$

Les dérivations $\Psi(D_2)$ et $\Psi(d_2)$ sont définies. De plus, dans $\overset{E_2}{\vdash} \Delta_2 = \Delta_1$, on retrouve les dérivations des jugements $\vdash \Gamma_2 = \Gamma_1$ et $\Gamma_1 \vdash K_2 = K_1$.

A partir des ces deux égalités et des égalités suivantes $\overset{E_5}{\vdash} \Gamma_s = \Gamma_1$ et $\Gamma_1 \overset{E_6}{\vdash} K_s = K_1$; il est possible d'obtenir des dérivations de E_7 et E_8. $\Psi(D_2')$ est donc définie.

Considérons maintenant

$$D_3' \equiv \quad \dfrac{\Gamma, x : K, \Gamma' \overset{D_3}{\vdash} S <_c Q \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]S <_{[k/x]c} [k/x]Q}\text{(SK.9)}$$

on a $\Psi(D_3') \equiv$

$$\dfrac{\Gamma_3, x : K_3, \Gamma_3' \overset{\Psi(D_3)}{\vdash} S_3 <_{c_3} Q_3 \qquad \dfrac{\dfrac{\Gamma_s \overset{\Psi(d_2)}{\vdash} k_s : K_s \qquad \overset{E_9}{\vdash} \Gamma_s = \Gamma_3}{\Gamma_3 \vdash k_s : K_s}\text{(3.3)} \qquad \Gamma_3 \overset{E_{10}}{\vdash} K_s = K_3}{\Gamma_3 \vdash k_s : K_3}\text{(3.1)}}{\Gamma_3, [k_s/x]\Gamma_3' \vdash [k_s/x]S_3 <_{[k_s/x]c_3} [k_s/x]Q_3}\text{(SK.9)}$$

Les dérivations $\Psi(D_3)$ et $\Psi(d_2)$ sont définies. De plus, dans $\overset{E_1}{\vdash} \Delta_3 = \Delta_1$, on retrouve les dérivations des jugements $\vdash \Gamma_3 = \Gamma_1$ et $\Gamma_1 \vdash K_3 = K_1$.

A partir des ces deux égalités et des égalités suivantes $\overset{E_5}{\vdash} \Gamma_s = \Gamma_1$ et $\Gamma_1 \overset{E_6}{\vdash} K_s = K_1$, il est possible d'obtenir des dérivations de E_7 et E_8. $\Psi(D_2')$ est donc définie.

Nous sommes en mesure de proposer une dérivations de $\Psi(d')$.

$$v_1 \equiv \dfrac{\Gamma_3, [k_s/x]\Gamma_3' \overset{\mathbf{co}^\circ(\Psi(D_3'))}{\vdash} [k_s/x]c_3 : ([k_s/x]S_3)[k_s/x]Q_3 \qquad \overset{?_1}{\vdash} \Gamma_3, [k_s/x]\Gamma_3' = \Gamma_1, [k_s/x]\Gamma_1'}{\Gamma_1, [k_s/x]\Gamma_1' \vdash [k_s/x]c_3 : ([k_s/x]S_3)[k_s/x]Q_3}\text{(3.3)}$$

$$v_2 \equiv \dfrac{\dfrac{\Gamma_2, [k_s/x]\Gamma_2' \overset{\Psi(D_2')}{\vdash} [k_s/x]q_2 : [k_s/x]S_2 \qquad \overset{?_2}{\vdash} \Gamma_2, [k_s/x]\Gamma_2' = \Gamma_1, [k_s/x]\Gamma_1'}{\Gamma_1, [k_s/x]\Gamma_1' \vdash [k_s/x]q_2 : [k_s/x]S_2}\text{(3.3)} \qquad \Gamma_1, [k_s/x]\Gamma_1' \overset{?_3}{\vdash} [k_s/x]S_2 = [k_s/x]S_3}{\Gamma_1, [k_s/x]\Gamma_1' \vdash [k_s/x]q_2 : [k_s/x]S_3}\text{(3.1)}$$

$$v_3 \equiv \dfrac{\Gamma_1, [k_s/x]\Gamma_1' \overset{v_1}{\vdash} [k_s/x]c_3 : ([k_s/x]S_3)[k_s/x]Q_3 \qquad \Gamma_1, [k_s/x]\Gamma_1' \overset{v_2}{\vdash} [k_s/x]q_2 : [k_s/x]S_3}{\Gamma_1, [k_s/x]\Gamma_1' \vdash [k_s/x]c_3([k_s/x]q_2) : [k_s/x]Q_3}\text{(5.5)}$$

$$\Psi(d') \equiv \cfrac{\Gamma_1, [k_s/x]\Gamma'_1 \overset{\Psi(D'_1)}{\vdash} [k_s/x]f_1 : (y : [k_s/x]Q_1)[k_s/x]Q'_1 \qquad \cfrac{\Gamma_1, [k_s/x]\Gamma'_1 \overset{v_3}{\vdash} [k_s/x]c_3([k_s/x]q_2) : [k_s/x]Q_3 \qquad \Gamma_1, [k_s/x]\Gamma'_1 \overset{?_4}{\vdash} [k_s/x]Q_3 = [k_s/x]Q_1}{\Gamma_1, [k_s/x]\Gamma'_1 \vdash [k_s/x]c_3([k_s/x]q_2) : [k_s/x]Q_1}\ (3.1)}{\Gamma_1, [k_s/x]\Gamma'_1 \vdash [k_s/x]f_1([k_s/x]c_3([k_s/x]q_2)) : [[k_s/x]c_3([k_s/x]q_2)/y][k_s/x]Q'_1}\ (5.5)$$

Démontrons que les différentes égalités $?_1$, $?_2$, $?_3$, $?_4$ sont dérivables.
L'égalité $\overset{?_1}{\vdash} \Gamma_3, [k_s/x]\Gamma'_3 = \Gamma_1, [k_s/x]\Gamma'_1$ est obtenue par autant de substitutions que nécessaires à partir des jugements $\overset{E_1}{\vdash} \Gamma_3, x : K_3, \Gamma'_3 = \Gamma_1, x : K_1, \Gamma'_1$, et $\Gamma_1 \vdash k_s : K_1$ que l'on retrouve dans $\Psi(D'_1)$.
L'égalité $\overset{?_2}{\vdash} \Gamma_2, [k_s/x]\Gamma'_2 = \Gamma_1, [k_s/x]\Gamma'_1$ est obtenue par le même procédé que $?_1$ mais à partir des jugements $\overset{E_2}{\vdash} \Gamma_2, x : K_2, \Gamma'_2 = \Gamma_1, x : K_1, \Gamma'_1$, et $\Gamma_1 \vdash k_s : K_1$.
L'égalité $\Gamma_1, [k_s/x]\Gamma'_1 \overset{?_3}{\vdash} [k_s/x]S_2 = [k_s/x]S_3$ est obtenue par substitution (règle 6.4) à partir des jugements $\Gamma_1, x : K_1, \Gamma'_1 \overset{E_3}{\vdash} S_2 = S_3$, et $\Gamma_1 \vdash k_s : K_1$.
L'égalité $\Gamma_1, [k_s/x]\Gamma'_1 \overset{?_4}{\vdash} [k_s/x]Q_3 = [k_s/x]Q_1$ est obtenue par le même procédé que $?_3$ mais à partir des jugements $\Gamma_1, x : K_1, \Gamma'_1 \overset{E_4}{\vdash} Q_3 = Q_1$, et $\Gamma_1 \vdash k_s : K_1$.
En conclusion $\Psi(d')$ est définie. De $\Psi(d') \sim \Psi(d)$, il suffit de comparer syntaxiquement les jugements finaux de $\Psi(d')$ et $\Psi(d)$.
En appliquant l'hypothèse d'induction et à l'aide du lemme 4.40, nous pouvons éliminer l'application des règles de substitutions des prémisses de d' et prouver que $\Psi(d) \sim \Psi(\mathbf{E}^\circ(d))$.

– Cas où d_1 se termine par 1.2.

$$d \equiv \cfrac{\cfrac{\Gamma \overset{d_0}{\vdash} K\mathbf{kind}}{\Gamma, x : K \vdash \mathbf{valid}}\ (1.2) \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma \vdash \mathbf{valid}}\ (6.1)$$

Par définition $\mathbf{E}^\circ(d) \equiv V^\circ_{\Gamma_1}(d_0)$. Grâce au lemme 4.40 et 4.41, on sait que $\Psi(V^\circ_{\Gamma_1}(d_0))$ est définie et $\Psi_d(\Gamma \vdash \mathbf{valid}) = \Psi_{d_0}(\Gamma \vdash \mathbf{valid}) = \Psi_{V^\circ_{\Gamma_1}(d_0)}(\Gamma \vdash \mathbf{valid})$. D'où $\Psi(\mathbf{E}^\circ(d)) \sim \Psi(d)$.

– Cas où d_1 se termine par 1.3.

$$d \equiv \cfrac{\cfrac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K}\,(1.3) \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash k : K}\,(6.5)$$

Prenons $d' \equiv$

$$\cfrac{\Gamma \overset{d_2}{\vdash} k : K \qquad \cfrac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid} \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash \textbf{valid}}\,(6.1)}{\Gamma, [k/x]\Gamma' \vdash k : K}\,(\text{wkn})$$

Le fait que $\Psi(d')$ soit définie et que $\Psi(d) \sim \Psi(d')$, est vérifié directement à partir de la définition de Ψ. Soit d'' obtenue en éliminant la règle 6.1 de la dérivation de la prémisse droite de d'. Par hypothèse d'induction, on a :

$$\Psi(\cfrac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid} \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash \textbf{valid}}\,(6.1)) \quad \sim \quad \Psi(\mathbf{E}_{sub}(\cfrac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid} \qquad \Gamma \overset{d_2}{\vdash} k : K}{\Gamma, [k/x]\Gamma' \vdash \textbf{valid}}\,(6.1)))$$

Grâce au lemme 4.40, on a $\Psi(d'') \sim \Psi(d')$, de plus $\mathbf{E}^\circ(d) \equiv \mathbf{E}^\circ_{wkn}(d'')$. On utilise le fait que la preuve est faite pour le cas de wkn pour montrer que $\Psi(d'') \sim \Psi(\mathbf{E}^\circ_{wkn}(d''))$. On peut conclure que $\Psi(d) \sim \Psi(\mathbf{E}^\circ(d))$.

□

Lemme 4.43. *Soit d une dérivation du jugement $\Gamma \vdash A <_c B :$ **Type** (ou $\Gamma \vdash K <_c K'$) dans $UTT^r[R]^-_w$. Si $\Psi(d)$ est définie alors :*

(a) $\Psi(\boldsymbol{l}^\circ_<(d))$ est définie et $\Psi_{\boldsymbol{l}^\circ_<(d)}(\Gamma \vdash A : \textbf{\textit{Type}}) = \Psi_d(\Gamma \vdash A : \textbf{\textit{Type}})$ (ou $\Psi_{\boldsymbol{l}^\circ_<(d)}(\Gamma \vdash K\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K\textbf{\textit{kind}})$),

(b) $\Psi(\boldsymbol{r}^\circ_<(d))$ est définie et $\Psi_{\boldsymbol{r}^\circ_<(d)}(\Gamma \vdash B : \textbf{\textit{Type}}) = \Psi_d(\Gamma \vdash B : \textbf{\textit{Type}}))$ (ou $\Psi_{\boldsymbol{r}^\circ_<(d)}(\Gamma \vdash K'\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K'\textbf{\textit{kind}})$),

(c) $\Psi(\boldsymbol{co}^\circ(d))$ *est définie et* $\Psi_{\boldsymbol{co}^\circ(d)}(\Gamma \vdash c : (El(A))El(B)) = \Psi_d(\Gamma \vdash c : (El(A))El(B))$
(ou $\Psi_{\boldsymbol{co}^\circ(d)}(\Gamma \vdash c : (K)K') = \Psi_d(\Gamma \vdash c : (K)K')$*).*

Démonstration. Preuve par induction sur la taille de d. Les règles à considérer sont toutes les règles de sous-typage (ST) et de sous-sorte (SK).

Cas de base. Le cas de base est couvert par la règle ST.1.

$$\text{d}\equiv \frac{\Gamma \overset{d_1}{\vdash} A : \textbf{Type} \qquad \Gamma \overset{d_2}{\vdash} B : \textbf{Type} \qquad \Gamma \overset{d_3}{\vdash} c : (El(A))El(B) \quad (A, c, B) \in C}{\Gamma \vdash A <_c B : \textbf{Type}} \text{ ST.1}$$

- $\mathbf{l}^\circ_<(d) \equiv d_1$. Par hypothèse $\Psi(d)$ est définie donc $\Psi(d_1)$ l'est également (lemme 4.40). On peut déduire que $\Psi(\mathbf{l}^\circ_<(d))$ est définie.
  On a $\Psi_{\mathbf{l}^\circ_<(d)}(\Gamma \vdash A : \textbf{Type}) = \Psi_{d_1}(\Gamma \vdash A : \textbf{Type})$, de plus on sait que $\Psi_d(\Gamma \vdash A : \textbf{Type}) = \Psi_{d_1}(\Gamma \vdash A : \textbf{Type})$, d'où $\Psi_{\mathbf{l}^\circ_<(d)}(\Gamma \vdash A : \textbf{Type}) = \Psi_d(\Gamma \vdash A : \textbf{Type})$.

- $\mathbf{r}^\circ_<(d) \equiv d_2$. Par une démonstration analogue, on montre que $\Psi_{\mathbf{r}^\circ_<(d)}(\Gamma \vdash B : \textbf{Type}) = \Psi_d(\Gamma \vdash B : \textbf{Type})$.

- $\mathbf{co}^\circ(d) \equiv d_3$. Par une démonstration analogue, on montre que $\Psi_{\mathbf{co}^\circ(d)}(\Gamma \vdash c : (El(A))El(B)) = \Psi_d(\Gamma \vdash c : (El(A))El(B))$.

Pas d'induction.
- Prenons le cas :

$$d \equiv \text{Cas } d \equiv \frac{\Gamma \overset{d_1}{\vdash} Q' <_{c'} Q \qquad \Gamma, x' : Q' \overset{d_2}{\vdash} [c'(x')/x]K <_{c''} K' \qquad \Gamma, x : Q \overset{d_3}{\vdash} K\textbf{kind}}{\Gamma \vdash (x : Q)K <_c (x' : Q')K'} \text{ (SK.4)}$$

avec $c \equiv [f : (x : Q)K][x' : Q']c''(f(c'(x')))$;

(a) on a $\mathbf{l}^\circ_<(d) \equiv \dfrac{\Gamma, x : Q \overset{d_3}{\vdash} K\textbf{kind}}{\Gamma \vdash (x : Q)K\textbf{kind}} \text{ (5.1)}$,
$\Psi(d_3)$ étant définie (lemme 4.40), par définition $\Psi(\mathbf{l}^\circ_<(d))$ l'est également. On a donc :

$\Psi_d(\Gamma \vdash (x:Q)K\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash (x:Q)K\mathbf{kind})$, de plus
$\Psi_{\mathbf{l}^\circ_<(d)}(\Gamma \vdash (x:Q)K\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash (x:Q)K\mathbf{kind})$, d'où
$\Psi_d(\Gamma \vdash (x:Q)K\mathbf{kind}) = \Psi_{\mathbf{l}^\circ_<(d)}(\Gamma \vdash (x:Q)K\mathbf{kind})$.

(b) on a $\mathbf{r}^\circ_<(d) \equiv \dfrac{\Gamma, x':Q' \overset{\mathbf{r}^\circ_<(d_2)}{\vdash} K'\mathbf{kind}}{\Gamma \vdash (x':Q')K'\mathbf{kind}}\,(5.1)$,
Par hypothèse d'induction $\Psi(\mathbf{r}^\circ_<(d_2))$ est définie, par définition $\Psi(\mathbf{r}^\circ_<(d))$ l'est également. On a donc :
$\Psi_d(\Gamma \vdash (x':Q')K'\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash (x':Q')K'\mathbf{kind})$, de plus
$\Psi_{\mathbf{r}^\circ_<(d)}(\Gamma \vdash (x':Q')K'\mathbf{kind}) = \Psi_{\mathbf{r}^\circ_<(d_2)}(\Gamma \vdash (x':Q')K'\mathbf{kind})$, or par hypothèse d'induction nous avons
$\Psi_{\mathbf{r}^\circ_<(d_2)}(\Gamma \vdash (x':Q')K'\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash (x':Q')K'\mathbf{kind})$ d'où
$\Psi_{\mathbf{r}^\circ_<(d)}(\Gamma \vdash (x':Q')K'\mathbf{kind}) = \Psi_d(\Gamma \vdash (x':Q')K'\mathbf{kind})$.

(c) Tout d'abord cherchons une dérivation du jugement :
$\Gamma \overset{\mathbf{co}^\circ(d)}{\vdash} c : ((x:Q)K)(x':Q')K'$

Considérons les dérivations suivantes :

$$p'_0 \equiv \dfrac{\dfrac{\Gamma, x:Q \overset{d_3}{\vdash} K\mathbf{kind}}{\Gamma \vdash (x:Q)K\mathbf{kind}}\,(5.1)}{\Gamma, f:(x:Q)K \vdash \mathbf{valid}}\,(1.2)$$

$$p_0 \equiv \dfrac{\Gamma, x':Q' \overset{V^\circ_{\Gamma_1}(d_2)}{\vdash} \mathbf{valid} \qquad \Gamma, f:(x:Q)K \overset{p'_0}{\vdash} \mathbf{valid}}{\Gamma, f:(x:Q)K, x':Q' \vdash \mathbf{valid}}\,(\text{wkn})$$

$$p_1 \equiv \dfrac{\Gamma, f:(x:Q)K, x':Q' \overset{\mathbf{E}^\circ(p_0)}{\vdash} \mathbf{valid}}{\Gamma, f:(x:Q)K, x':Q' \vdash x':Q'}\,(1.3)$$

$$p_2 \equiv \dfrac{\Gamma, f:(x:Q)K, x':Q' \overset{\mathbf{E}^\circ(p_0)}{\vdash} \mathbf{valid}}{\Gamma, f:(x:Q)K, x':Q' \vdash f:(x:Q)K}\,(1.3)$$

$$h \equiv \dfrac{\Gamma, x':Q' \overset{\mathbf{co}^\circ(d_2)}{\vdash} c'' : ([c'(x')/x]K)K' \qquad \dfrac{\dfrac{\Gamma, x:Q \overset{d_3}{\vdash} K\mathbf{kind}}{\Gamma \vdash (x:Q)K\mathbf{kind}}\,(5.1)}{\Gamma, f:(x:Q)K \vdash \mathbf{valid}}\,(1.2)}{\Gamma, f:(x:Q)K, x':Q' \vdash c'' : ([c'(x')/x]K)K'}\,(\text{wkn})$$

$$h_1 \equiv \frac{\Gamma \overset{\mathbf{co}^\circ(d_1)}{\vdash} c' : (Q')Q \qquad \Gamma, f : (x : Q)K, x' : Q' \overset{\mathbf{E}^\circ(p_0)}{\vdash} \mathbf{valid}}{\Gamma, f : (x : Q)K, x' : Q' \vdash c' : (Q')Q} \text{(wkn)}$$

$$h_2 \equiv \frac{\Gamma, f : (x : Q)K, x' : Q' \overset{\mathbf{E}^\circ(h_1)}{\vdash} c' : (Q')Q \qquad \Gamma, f : (x : Q)K, x' : Q' \overset{p_1}{\vdash} x' : Q'}{\Gamma, f : (x : Q)K, x' : Q' \vdash c'(x') : Q} \text{(5.5)}$$

$$h_3 \equiv \frac{\Gamma, f : (x : Q)K, x' : Q' \overset{h_2}{\vdash} c'(x') : Q \qquad \Gamma, f : (x : Q)K, x' : Q' \overset{p_2}{\vdash} f : (x : Q)K}{\Gamma, f : (x : Q)K, x' : Q' \vdash f(c'(x')) : [c'(x')/x]K} \text{(5.5)}$$

Et $h_4 \equiv$

$$\frac{\Gamma, f : (x : Q)K, x' : Q' \overset{\mathbf{E}^\circ(h)}{\vdash} c'' : ([c'(x')/x]K)K' \qquad \Gamma, f : (x : Q)K, x' : Q' \overset{h_3}{\vdash} f(c'(x')) : [c'(x')/x]K}{\Gamma, f : (x : Q)K, x' : Q' \vdash c''(f(c'(x'))) : K'} \text{(5.5)}$$

On obtient finalement la dérivation :

$$\mathbf{co}^\circ(d) \equiv \frac{\dfrac{\Gamma, f : (x : Q)K, x' : Q' \overset{h_4}{\vdash} c''(f(c'(x'))) : K'}{\Gamma, f : (x : Q)K \vdash [x' : Q']c''(f(c'(x'))) : (x' : Q')K'} \text{(5.3)}}{\Gamma \vdash [f : (x : Q)K][x' : Q']c''(f(c'(x'))) : (f : (x : Q)K)(x' : Q')K'} \text{(5.3)}$$

Montrons que $\Psi(\mathbf{co}^\circ(d))$ est définie.
Sachant que $\Psi(d_3)$ est définie, par définition $\Psi(p_0')$ l'est également. Voici ci-dessous une dérivation de $\Psi(p_0)$.

$$\Psi(p_0) \equiv \frac{\Psi_{V^\circ_{\Gamma_1}(d_2)}(\Gamma), x' : \Psi_{V^\circ_{\Gamma_1}(d_2)}(Q') \overset{\Psi(V^\circ_{\Gamma_1}(d_2))}{\vdash} \mathbf{valid} \qquad \dfrac{\Psi_{d_3}(\Gamma), f : (x : \Psi_{d_3}(Q))\Psi_{d_3}(K) \overset{\Psi(p_0')}{\vdash} \mathbf{valid} \qquad \overset{E_1}{\vdash} \Psi_{d_3}(\Gamma) = \Psi_{V^\circ_{\Gamma_1}(d_2)}(\Gamma)}{\Psi_{V^\circ_{\Gamma_1}(d_2)}(\Gamma), f : (x : \Psi_{d_3}(Q))\Psi_{d_3}(K) \vdash \mathbf{valid}} \text{(3.3)}}{\Psi_{V^\circ_{\Gamma_1}(d_2)}(\Gamma), f : (x : \Psi_{d_3}(Q)), x' : \Psi_{V^\circ_{\Gamma_1}(d_2)}(Q') \vdash \mathbf{valid}} \text{(wkn)}$$

Grâce au lemme 4.41 on sait que $\Psi_{V^\circ_{\Gamma_1}(d_2)}(\Gamma \vdash \mathbf{valid}) = \Psi_{d_2}(\Gamma \vdash \mathbf{valid})$. De plus, par définition de $\Psi(d)$, on a $\Psi_{d_3}(\Gamma \vdash \mathbf{valid}) = \Psi_{d_2}(\Gamma \vdash \mathbf{valid})$. Le jugement $\overset{E_1}{\vdash} \Psi_{d_3}(\Gamma) = \Psi_{V^\circ_{\Gamma_1}(d_2)}(\Gamma)$ est donc dérivable et $\Psi(p_0)$ est définie.
Nous savons par rapport au lemme 4.42 que $\Psi(\mathbf{E}^\circ(p_0))$ est définie, par définition $\Psi(p_1)$ et $\Psi(p_2)$ le sont également. Par une démonstration analogue à celle de $\Psi(p_0)$, nous prouvons que $\Psi(h)$ et $\Psi(h_1)$ sont

définies.

Pour prouver que $\Psi(h_2)$ est définie, selon sa définition nous devons montrer l'égalité :
$\Psi_{\mathbf{E}^\circ(h_1)}(\Gamma, f : (x : Q)K, x' : Q' \vdash \textbf{valid}) = \Psi_{p_1}(\Gamma, f : (x : Q)K, x' : Q' \vdash \textbf{valid})$.
Par hypothèse d'induction, par définition de Ψ et grâce au lemme 4.42, nous sommes mesure de prouver les deux égalités :
$\Psi_{\mathbf{E}^\circ(h_1)}(\Gamma, f : (x : Q)K, x' : Q' \vdash \textbf{valid}) = \Psi_{p_0}(\Gamma, f : (x : Q)K, x' : Q' \vdash \textbf{valid})$, et
$\Psi_{p_1}(\Gamma, f : (x : Q)K, x' : Q' \vdash \textbf{valid}) = \Psi_{p_0}(\Gamma, f : (x : Q)K, x' : Q' \vdash \textbf{valid})$,
et en déduire l'égalité que nous recherchons à savoir :
$\Psi_{\mathbf{E}^\circ(h_1)}(\Gamma, f : (x : Q)K, x' : Q' \vdash \textbf{valid}) = \Psi_{p_1}(\Gamma, f : (x : Q)K, x' : Q' \vdash \textbf{valid})$.

Par une démonstration analogue, nous prouvons que $\Psi(h_3)$ et $\Psi(h_4)$ sont définies. Nous pouvons conclure en disant que $\Psi(\mathbf{co}^\circ(d))$ est définie.

Montrons maintenant que :
$\Psi_{\mathbf{co}^\circ(d)}(\Gamma \vdash [f : (x : Q)K][x' : Q']c''(f(c'(x') : ((x : Q)K)(x' : Q')K') = \Psi_d(\Gamma \vdash [f : (x : Q)K][x' : Q']c''(f(c'(x') : ((x : Q)K)(x' : Q')K')$.

En analysant la dérivation $\Psi(d)$ on peut dire que :
$\Psi_d(\Gamma \vdash [f : (x : Q)K][x' : Q']c''(f(c'(x') : ((x : Q)K)(x' : Q')K') =$
$\Psi_{d_1}(\Gamma) \vdash [f : (x : \Psi_{d_3}(Q))\Psi_{d_3}(K)][x' : \Psi_{d_3}(Q')]\Psi_{d_2}(c'')(f(\Psi_{d_1}(c')(x') : ((x : \Psi_{d_3}(Q))\Psi_{d_3}(K))(x' : \Psi_{d_3}(Q'))\Psi_{d_2}(K'))$.
De plus l'observation de **co**°(d) et le lemme 4.42 nous permettent d'affirmer que :
$\Psi_{\mathbf{co}^\circ(d)}(\Gamma \vdash [f : (x : Q)K][x' : Q']c''(f(c'(x') : ((x : Q)K)(x' : Q')K') =$
$\Psi_{\mathbf{co}^\circ(d_1)}(\Gamma) \vdash [f : (x : \Psi_{d_3}(Q))\Psi_{d_3}(K)][x' : \Psi_{d_3}(Q')]\Psi_{\mathbf{co}^\circ(d_2)}(c'')(f(\Psi_{\mathbf{co}^\circ(d_1)}(c')(x') : ((x : \Psi_{d_3}(Q))\Psi_{d_3}(K))(x' : \Psi_{d_3}(Q'))\Psi_{\mathbf{co}^\circ(d_2)}(K'))$.
Par hypothèse d'induction, nous avons les égalités suivantes :
$\Psi_{\mathbf{co}^\circ(d_1)}(\Gamma \vdash c' : (Q')Q) = \Psi_{d_1}(\Gamma \vdash c' : (Q')Q)$ et

$\Psi_{\mathbf{co}^\circ(d_2)}(\Gamma \vdash c'' : ([c'(x')/x]K)K') = \Psi_{d_2}(\Gamma \vdash c'' : ([c'(x')/x]K)K')$.
De ces deux égalités, nous pouvons déduire les égalités suivantes :
$\Psi_{\mathbf{co}^\circ(d_1)}(\Gamma) = \Psi_{d_1}(\Gamma), \quad \Psi_{\mathbf{co}^\circ(d_2)}(c'') = \Psi_{d_2}(c''), \quad \Psi_{\mathbf{co}^\circ(d_1)}(c') = \Psi_{d_1}(c')$,
$\Psi_{\mathbf{co}^\circ(d_2)}(K') = \Psi_{d_2}(K')$
On peut en conclure que :
$\Psi_{\mathbf{co}^\circ(d)}(\Gamma \vdash [f : (x : Q)K][x' : Q']c''(f(c'(x') : ((x : Q)K)(x' : Q')K') =$
$\Psi_d(\Gamma \vdash [f : (x : Q)K][x' : Q']c''(f(c'(x') : ((x : Q)K)(x' : Q')K')$.

□

Lemme 4.44. *Soit d une $UTT^r[R]_w^-$-dérivation d'un des jugements $\Gamma \vdash K_1 = K_2$, $\Gamma \vdash k_1 = k_2 : K$ ou $\Gamma \vdash \Sigma : K$ (Σ dénote un terme ou une égalité de termes). Si $\Psi(d)$ est définie alors $\Psi(\boldsymbol{l}^\circ_{\equiv}(d))$ et $\Psi(\boldsymbol{r}^\circ_{\equiv}(d))$ sont définies, $\Psi(l^\circ_{\equiv}(d))$ et $\Psi(r^\circ_{\equiv}(d))$ sont définies, et $\Psi(\boldsymbol{kd}^\circ(d))$ est définie respectivement.*
De plus :

(a) $\Psi_{\boldsymbol{l}^\circ_{\equiv}(d)}(\Gamma \vdash K_1\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K_1\textbf{\textit{kind}}), \quad \Psi_{\boldsymbol{r}^\circ_{\equiv}(d)}(\Gamma \vdash K_2\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K_2\textbf{\textit{kind}})$ *;*

(b) $\Psi_{l^\circ_{\equiv}(d)}(\Gamma \vdash k_1 : K) = \Psi_d(\Gamma \vdash k_1 : K), \quad \Psi_{r^\circ_{\equiv}(d)}(\Gamma \vdash k_2 : K) = \Psi_d(\Gamma \vdash k_2 : K)$ *;*

(c) $\Psi_{\boldsymbol{kd}^\circ(d)}(\Gamma \vdash K\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K\textbf{\textit{kind}})$.

Démonstration. Par induction sur la taille de d.
Cas de base. Quand $\Psi(d) \equiv d$.

Pas d'induction.

(a) Le jugement ayant la forme $\Gamma \vdash K_1 = K_2$, la dérivation d peut se terminer par une des règles suivantes : 2.1, 2.2, 2.3, 4.3, 5.2.

- Considérons le cas :

$$d \equiv \quad \frac{\Gamma, x : K_1 \overset{d_1}{\vdash} K_1' = K_2' \qquad \Gamma \overset{d_2}{\vdash} K_1 = K_2}{\Gamma \vdash (x : K_1)K_1' = (x : K_2)K_2'} \text{(5.2)}$$

– On a $\mathbf{l}^\circ_=(d) \equiv \dfrac{\Gamma, x : K_1 \overset{\mathbf{l}^\circ_=(d_1)}{\vdash} K_1'\mathbf{kind}}{\Gamma \vdash (x : K_1)K_1'\mathbf{kind}}$ (5.1)

Par hypothèse d'induction $\Psi(\mathbf{l}^\circ_=(d_1))$ est définie, il en est donc de même pour $\Psi(\mathbf{l}^\circ_=(d))$.
Par définition de Ψ on a :
$\Psi_d(\Gamma \vdash (x : K_1)K_1'\mathbf{kind}) = \Psi_{d_1}(\Gamma) \vdash (x : \Psi_{d_1}(K_1))\Psi_{d_1}(K_1')\mathbf{kind})$,
de plus $\Psi_{\mathbf{l}^\circ_=(d)}(\Gamma \vdash (x : K_1)K_1'\mathbf{kind}) = \Psi_{\mathbf{l}^\circ_=(d_1)}(\Gamma) \vdash (x : \Psi_{\mathbf{l}^\circ_=(d_1)}(K_1))\Psi_{\mathbf{l}^\circ_=(d_1)}(K_1')\mathbf{kind})$,
or par hypothèse d'induction on a l'égalité suivante :
$\Psi_{\mathbf{l}^\circ_=(d_1)}(\Gamma, x : K_1 \vdash K_1'\mathbf{kind}) = \Psi_{d_1}(\Gamma, x : K_1 \vdash K_1'\mathbf{kind})$, on peut en déduire les égalités :
$\Psi_{\mathbf{l}^\circ_=(d_1)}(\Gamma) = \Psi_{d_1}(\Gamma), \quad \Psi_{\mathbf{l}^\circ_=(d_1)}(K_1) = \Psi_{d_1}(K_1), \quad \Psi_{\mathbf{l}^\circ_=(d_1)}(K_1') = \Psi_{d_1}(K_1')$,
d'où
$\Psi_{\mathbf{l}^\circ_=(d)}(\Gamma \vdash (x : K_1)K_1'\mathbf{kind}) = \Psi_d(\Gamma \vdash (x : K_1)K_1'\mathbf{kind})$.

– on a $\mathbf{r}^\circ_=(d) \equiv$

$$\mathbf{E}^\circ(\frac{\Gamma, x : K_1 \overset{\mathbf{r}^\circ_=(d_1)}{\vdash} K_2'\mathbf{kind} \qquad \Gamma \overset{d_2}{\vdash} K_1 = K_2 \qquad \Gamma \overset{\mathbf{r}^\circ_=(d_2)}{\vdash} K_2\mathbf{kind}}{\dfrac{\Gamma, x : K_2 \vdash K_2'\mathbf{kind}}{\Gamma \vdash (x : K_2)K_2'\mathbf{kind}} \text{(5.1)}} \text{(3.3')}$$

Montrons tout d'abord que $\Psi(\mathbf{r}^\circ_=(d))$ est définie.
Nous savons que $\Psi(d_2)$ est définie, il en est de même pour $\Psi(\mathbf{r}^\circ_=(d_1))$ et $\Psi(\mathbf{r}^\circ_=(d_2))$ par hypothèse d'induction. Connaissant la définition de $\Psi(\mathbf{r}^\circ_=(d))$ et en appliquant le lemme 4.42, nous savons qu'il nous suffit de prouver les égalités :
$\Psi_{\mathbf{r}^\circ_=(d_1)}(\Gamma \vdash K_1\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K_1\mathbf{kind})$ et $\Psi_{d_2}(\Gamma \vdash K_2\mathbf{kind}) = \Psi_{\mathbf{r}^\circ_=(d_2)}(\Gamma \vdash K_2\mathbf{kind})$.
La seconde égalité découle directement des hypothèses d'induction.
Penchons nous sur la première. Par définition de $\Psi(d)$, on sait que $\Psi_{d_1}(\Gamma \vdash K_1\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K_1\mathbf{kind})$ et par hypothèse d'induction on sait que $\Psi_{\mathbf{r}^\circ_=(d_1)}(\Gamma \vdash K_1\mathbf{kind}) = \Psi_{d_1}(\Gamma \vdash K_1\mathbf{kind})$. De ces deux égalités, on en déduit celle que nous recherchons, à savoir $\Psi_{\mathbf{r}^\circ_=(d_1)}(\Gamma \vdash K_1\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K_1\mathbf{kind})$. On peut conclure que $\Psi(\mathbf{r}^\circ_=(d))$ est définie.

Nous devons maintenant montrer que :
$\Psi_{\mathbf{r}^\circ_=(d)}(\Gamma \vdash (x:K_2)K_2'\mathbf{kind}) = \Psi_d(\Gamma \vdash (x:K_2)K_2'\mathbf{kind})$.
Par définition de Ψ on a :
$\Psi_d(\Gamma \vdash (x:K_2)K_2'\mathbf{kind}) = \Psi_{d_2}(\Gamma) \vdash (x:\Psi_{d_2}(K_2))\Psi_{d_1}(K_2')\mathbf{kind}$, de plus le lemme 4.42 nous permet d'affirmer également que :
$\Psi_{\mathbf{r}^\circ_=(d)}(\Gamma \vdash (x:K_2)K_2'\mathbf{kind}) = \Psi_{d_2}(\Gamma) \vdash (x:\Psi_{d_2}(K_2))\Psi_{\mathbf{r}^\circ_=(d_1)}(K_2')\mathbf{kind}$.
Par hypothèse d'induction on a l'égalité suivante :
$\Psi_{\mathbf{r}^\circ_=(d_1)}(\Gamma, x:K_1 \vdash K_2'\mathbf{kind}) = \Psi_{d_1}(\Gamma, x:K_1 \vdash K_2'\mathbf{kind})$, on peut en déduire l' égalité : $\Psi_{\mathbf{r}^\circ_=(d_1)}(K_2') = \Psi_{d_1}(K_2')$, d'où
$\Psi_{\mathbf{r}^\circ_=(d)}(\Gamma \vdash (x:K_2)K_2'\mathbf{kind}) = \Psi_d(\Gamma \vdash (x:K_2)K_2'\mathbf{kind})$.

(b) Le jugement ayant la forme $\Gamma \vdash k_1 = k_2 : K$, la dérivation d peut se terminer par une des règles suivantes : 2.5, 2.6, 3.2, 5.4, 5.6, 5.7, 5.8, CA.2, CD ou encore une des règles d'égalité associées aux constantes de $UTT^r[R]^-_w$.

- Considérons le cas

$$d \equiv \quad \frac{\Gamma \overset{d_1}{\vdash} f = f' : (x:K)K' \qquad \Gamma \overset{d_2}{\vdash} k_1 = k_2 : K_0 \qquad \Gamma \overset{d_3}{\vdash} K_0 <_c K}{\Gamma \vdash f(k_1) = f'(k_2) : [c(k_1)/x]K'} \quad \text{(CA.2)}$$

  – on a $l^\circ_=(d) \equiv \dfrac{\Gamma \overset{l^\circ_=(d_1)}{\vdash} f : (x:K)K' \qquad \Gamma \overset{l^\circ_=(d_2)}{\vdash} k_1 : K_0 \qquad \Gamma \overset{d_3}{\vdash} K_0 <_c K}{\Gamma \vdash f(k_1) : [c(k_1)/x]K'}$ (CA.1)

  Montrons tout d'abord que $\Psi(l^\circ_=(d))$ est définie.
  Nous savons que $\Psi(l^\circ_=(d_1))$, $\Psi(l^\circ_=(d_2))$ et $\Psi(d_3)$ sont définies. Il nous suffit de prouver les égalités :
  $\Psi_{l^\circ_=(d_1)}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash K\mathbf{kind})$ et $\Psi_{l^\circ_=(d_2)}(\Gamma \vdash K_0\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash K_0\mathbf{kind})$.
  Par définition de $\Psi(d)$ on peut prouver que :
  $\Psi_{d_1}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash K\mathbf{kind})$ et $\Psi_{d_2}(\Gamma \vdash K_0\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash K_0\mathbf{kind})$.
  De même que l'on peut prouver à l'aide des hypothèses d'induction que :
  $\Psi_{l^\circ_=(d_1)}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_1}(\Gamma \vdash K\mathbf{kind})$ et $\Psi_{l^\circ_=(d_2)}(\Gamma \vdash K_0\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K_0\mathbf{kind})$.
  Nous pouvons en déduire les égalités recherchées et prouver que $\Psi(l^\circ_=(d))$ est définie.

Nous devons maintenant montrer que :
$\Psi_{l^\circ_=(d)}(\Gamma \vdash f(k_1) : [c(k_1)/x]K') = \Psi_d(\Gamma \vdash f(k_1) : [c(k_1)/x]K')$.
Ce résultat est facilement prouvable par définition de $\Psi(d)$ et de $\Psi(l^\circ_=(d))$ et à l'aide des hypothèses d'induction.

– On a $r^\circ_=(d) \equiv$

$$\dfrac{\dfrac{\Gamma \overset{r^\circ_=(d_1)}{\vdash} f' : (x:K)K' \qquad \Gamma \overset{r^\circ_=(d_2)}{\vdash} k_2 : K_0 \qquad \Gamma \overset{d_3}{\vdash} K_0 <_c K}{\Gamma \vdash f'(k_2) : [c(k_2)/x]K'}\,\text{(CA.1)} \qquad \dfrac{\Gamma \overset{\mathbf{E}^\circ(h)}{\vdash} [c(k_1)/x]K' = [c(k_2)/x]K'}{\Gamma \vdash [c(k_2)/x]K' = [c(k_1)/x]K'}\,\text{(2.2)}}{\Gamma \vdash f'(k_2) : [c(k_1)/x]K'}\,\text{(3.1)}$$

Avec $h \equiv$

$$\dfrac{\dfrac{p\_^\circ(\mathbf{kd}^\circ(d_1))}{\Gamma, x : K \vdash K'\mathbf{kind}} \qquad \dfrac{\dfrac{\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (K_0)K}{\Gamma, x : K \vdash c = c : (K_0)K}\,\text{(2.4)} \qquad \Gamma \overset{d_2}{\vdash} k_1 = k_2 : K_0}{\Gamma \vdash c(k_1) = c(k_2) : K}\,\text{(5.6)} \qquad \dfrac{\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (K_0)K \qquad \Gamma \overset{\mathbf{r}^\circ_<(d_2)}{\vdash} k_2 : K_0}{\Gamma \vdash c(k_2) : K}\,\text{(5.5)}}{\Gamma \vdash [c(k_1)/x]K' = [c(k_2)/x]K'}\,\text{(6.6')}$$

Par une méthode relativement similaire aux cas précédents (en employant également les lemmes 4.42 et 4.43), nous sommes en mesure de prouver que $\Psi(r^\circ_=(d))$ est définie et que :
$\Psi_{r^\circ_=(d)}(\Gamma \vdash f'(k_2) : [c(k_1)/x]K') = \Psi_d(\Gamma \vdash f'(k_2) : [c(k_1)/x]K')$.

(c) Le jugement ayant la forme $\Gamma \vdash \Sigma : K$, la dérivation d peut se terminer par une des règles suivantes : 3.1, 3.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, ST.1, CA.1, CA.2, CD ou encore une des règles associées aux constantes de $UTT^r[R]^-_w$.

- Considérons le cas

$$d \equiv \quad \dfrac{\Gamma \overset{d_1}{\vdash} f = f' : (x:K)K' \qquad \Gamma \overset{d_2}{\vdash} k_1 = k_2 : K_0 \qquad \Gamma \overset{d_3}{\vdash} K_0 <_c K}{\Gamma \vdash f(k_1) = f'(k_2) : [c(k_1)/x]K'}\,\text{(CA.2)}$$

on a :

$$\mathbf{kd}^\circ(d) \equiv \quad \mathbf{E}^\circ\left(\dfrac{\dfrac{p\_^\circ(\mathbf{kd}^\circ(d_1))}{\Gamma, x : K \vdash K'\mathbf{kind}} \qquad \dfrac{\Gamma \overset{\mathbf{co}^\circ(d_3)}{\vdash} c : (K_0)K \qquad \Gamma \overset{\mathbf{l}^\circ_<(d_2)}{\vdash} k_1 : K_0}{\Gamma \vdash c(k_1) : K}\,\text{(5.5)}}{\Gamma \vdash [c(k_1)/x]K'\mathbf{kind}}\,\text{(6.2)}\right)$$

Par une méthode relativement similaire aux cas précédents nous prouvons que $\Psi(\mathbf{kd}^\circ(d))$ est définie et que
$\Psi_{\mathbf{kd}^\circ(d)}(\Gamma \vdash [c(k_1)/x]K'\mathbf{kind}) = \Psi_d(\Gamma \vdash [c(k_1)/x]K'\mathbf{kind})$.

□

Lemme 4.45. *Soit d une $UTT^r[R]^-_w$-dérivation et* ***less*** *l'algorithme vu au lemme 4.21. Si $\Psi(d)$ est définie alors $\Psi(\textbf{less}(d))$ est également définie, de plus $\Psi(\textbf{less}(d)) \sim \Psi(d)$.*

Démonstration. Par induction sur le nombre de $UTT^r[R]^-_w$-règles qui ne sont pas des $UTT^r[R]^-$-règles.
Cas de base. Lorsque d ne contient aucune des règles mentionnées plus haut : évident.
Pas d'induction. Les règles à considérer sont les règles ST.2', ST.3', SK.5' et SK.6'.
Considérons le cas ST.2' dont la méthode est similaire pour tous les autres cas.

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} A <_c B : \textbf{Type} \qquad \Gamma \overset{d_2}{\vdash} A = A' : \textbf{Type} \quad \Gamma \overset{d_3}{\vdash} A' : \textbf{Type}}{\Gamma \vdash A' <_c B : \textbf{Type}} \text{(ST.2')}$$

On sait que $\textbf{less}(d) \equiv \dfrac{\Gamma \overset{\textbf{less}(d_1)}{\vdash} A <_c B : \textbf{Type} \qquad \Gamma \overset{\textbf{less}(d_2)}{\vdash} A = A' : \textbf{Type}}{\Gamma \vdash A' <_c B : \textbf{Type}} \text{(ST.2)}$

Nous prouvons d'abord que $\Psi(\textbf{less}(d))$ est définie, par une méthode similaire à celle employée lors des démontrations des lemmes 4.42, 4.43 et 4.44.

Montrons maintenant que $\Psi(\textbf{less}(d)) \sim \Psi(d)$, autrement dit que
$\Psi_d(\Gamma \vdash A' <_c B : \textbf{Type}) = \Psi_{\textbf{less}(d)}(\Gamma \vdash A' <_c B : \textbf{Type})$.
Par définition de $\Psi(d)$ on sait que :
(1) $\Psi_d(\Gamma \vdash A' <_c B : \textbf{Type}) = \Psi_{d_1}(\Gamma) \vdash \Psi_{d_2}(A') <_{\Psi_{d_1}(c)} \Psi_{d_1}(B)$. De plus par définition de $\Psi(\textbf{less}(d))$ on sait que :
(2)$\Psi_{\textbf{less}(d)}(\Gamma \vdash A' <_c B : \textbf{Type}) = \Psi_{\textbf{less}(d_1)}(\Gamma) \vdash \Psi_{\textbf{less}(d_2)}(A') <_{\Psi_{\textbf{less}(d_1)}(c)} \Psi_{\textbf{less}(d_1)}(B)$. En appliquant l'hypothèse d'induction à d_1 et d_2, on a $\Psi(\textbf{less}(d_1)) \sim \Psi(d_1)$ et $\Psi(\textbf{less}(d_2)) \sim \Psi(d_2)$. On peut conclure que $\Psi_d(\Gamma \vdash A' <_c B : \textbf{Type}) = \Psi_{\textbf{less}(d)}(\Gamma \vdash A' <_c B : \textbf{Type})$ donc que $\Psi(\textbf{less}(d)) \sim \Psi(d)$.

□

Lemme 4.46. *Soit d une $UTT^r[R]$-dérivation et $\boldsymbol{E}^*$ l'algorithme vu au lemme 4.22. Si $\Psi(d)$ est définie alors $\Psi(\boldsymbol{E}^*(d))$ est également définie, de plus $\Psi(\boldsymbol{E}^*(d)) \sim$*

$\Psi(d)$.

Démonstration. Preuve par induction sur le nombre de règles de $UTT^r[R]$ qui ne sont pas des règles de $UTT^r[R]^-_w$.
Cas de base. Il n'existe aucune des règles mentionnées plus haut dans d. Dans ce cas $\mathbf{E}^*(d) \equiv d$.

Pas d'induction.

- Cas où la dernière règle de d est une des règles dont il existe une version avec des prémisses supplémentaires.

  Exemple du cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} K_1 <_c K_2 \qquad \Gamma \overset{d_2}{\vdash} K_2 = K_2'}{\Gamma \vdash K_1 <_c K_2'}$ (SK.5)

  On sait que :
  $$\mathbf{E}^*(d) \equiv \frac{\Gamma \overset{\mathbf{E}^*(d_1)}{\vdash} K_1 <_c K_2 \qquad \Gamma \overset{\mathbf{E}^*(d_2)}{\vdash} K_2 = K_2' \quad \Gamma \overset{\mathbf{r}^{\circ}_{=}(\mathbf{E}^*(d_2))}{\vdash} K_2'\mathbf{kind}}{\Gamma \vdash K_1 <_c K_2'} \text{ (SK.5')}$$

  Il faut d'abord montrer que $\Psi(\mathbf{E}^*(d))$ est définie.
  Afin d'y parvenir, connaissant la définition de $\Psi(\mathbf{E}^*(d))$, nous savons qu'il nous suffit de prouver les égalités :
  $\Psi_{\mathbf{E}^*(d_1)}(\Gamma \vdash K_2\mathbf{kind}) = \Psi_{\mathbf{E}^*(d_2)}(\Gamma \vdash K_2\mathbf{kind})$ et $\Psi_{\mathbf{E}^*(d_2)}(\Gamma \vdash K_2'\mathbf{kind}) = \Psi_{\mathbf{r}^{\circ}_{=}(\mathbf{E}^*(d_2))}(\Gamma \vdash K_2'\mathbf{kind})$.
  Par définition de $\Psi(d)$, en appliquant les hypothèses d'induction et le lemme 4.44, nous sommes en mesure de prouver que les deux égalités précédentes sont dérivables et donc que $\Psi(\mathbf{E}^*(d))$ est définie.

  Montrons maintenant que $\Psi(\mathbf{E}^*(d)) \sim \Psi(d)$.
  Par définition de $\Psi(d)$, on sait que :
  (1) $\Psi_d(\Gamma \vdash K_1 <_c K_2') = \Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(K_1) <_{\Psi_{d_1}(c)} \Psi_{d_2}(K_2')$. De plus par définition de $\Psi(\mathbf{E}^*(d))$, on sait que :
  (2) $\Psi_{\mathbf{E}^*(d)}(\Gamma \vdash K_1 <_c K_2') = \Psi_{\mathbf{E}^*(d_1)}(\Gamma) \vdash \Psi_{\mathbf{E}^*(d_1)}(K_1) <_{\Psi_{\mathbf{E}^*(d_1)}(c)} \Psi_{\mathbf{E}^*(d_2)}(K_2')$.
  En appliquant l'hypothèse d'induction à d_1 et d_2, on a $\Psi(\mathbf{E}^*(d_1)) \sim \Psi(d_1)$ et $\Psi(\mathbf{E}^*(d_2)) \sim \Psi(d_2)$. On peut conclure que $\Psi_d(\Gamma \vdash K_1 <_c K_2') = \Psi_{\mathbf{E}^*(d)}(\Gamma \vdash K_1 <_c K_2')$ donc que $\Psi(\mathbf{E}^*(d)) \sim \Psi(d)$.

- Cas où la dernière règle de d est une des règles à éliminer.

Considérons le cas $d \equiv \dfrac{\Gamma_1, \Gamma_2 \overset{d_1}{\vdash} J \qquad \Gamma_1, \Gamma_3 \overset{d_2}{\vdash} \textbf{valid}}{\Gamma_1, \Gamma_3, \Gamma_2 \vdash J}$ (wkn)

On sait que :

$$\mathbf{E}^*(d) \equiv \mathbf{E}^\circ\left(\dfrac{\Gamma_1, \Gamma_2 \overset{\mathbf{E}^*(d_1)}{\vdash} J \qquad \Gamma_1, \Gamma_3 \overset{\mathbf{E}^*(d_2)}{\vdash} \textbf{valid}}{\Gamma_1, \Gamma_3, \Gamma_2 \vdash J} \text{(wkn)}\right)$$

Il faut d'abord montrer que $\Psi(\mathbf{E}^*(d))$ est définie.
Pour cela il suffit juste de prouver que :
$\Psi_{\mathbf{E}^*(d_1)}(\Gamma \vdash \textbf{valid}) = \Psi_{\mathbf{E}^*(d_2)}(\Gamma \vdash \textbf{valid})$.
Cette égalité est prouvée facilement par définition de $\Psi(d)$ et en appliquant les hypothèses d'induction. $\Psi(\mathbf{E}^*(d))$ est donc définie.

Montrons que $\Psi(\mathbf{E}^*(d)) \sim \Psi(d)$.
Par définition de $\Psi(d)$ on sait que :
(1) $\Psi_d(\Gamma_1, \Gamma_3, \Gamma_2 \vdash J) = \Psi_{d_1}(\Gamma_1), \Psi_{d_2}(\Gamma_3), \Psi_{d_1}(\Gamma_2) \vdash \Psi_{d_1}(J)$. De plus par définition de $\Psi(\mathbf{E}^*(d))$ on sait que :
(2) $\Psi_{\mathbf{E}^*(d)}(\Gamma_1, \Gamma_3, \Gamma_2 \vdash J) = \Psi_{\mathbf{E}^*(d_1)}(\Gamma_1), \Psi_{\mathbf{E}^*(d_2)}(\Gamma_3), \Psi_{\mathbf{E}^*(d_1)}(\Gamma_2) \vdash \Psi_{\mathbf{E}^*(d_1)}(J)$.
En appliquant l'hypothèse d'induction à d_1 et d_2, on a $\Psi(\mathbf{E}^*(d_1)) \sim \Psi(d_1)$ et $\Psi(\mathbf{E}^*(d_2)) \sim \Psi(d_2)$ et le lemme 4.42 nous permet de conclure que $\Psi_d(\Gamma_1, \Gamma_3, \Gamma_2 \vdash J) = \Psi_{\mathbf{E}^*(d)}(\Gamma_1, \Gamma_3, \Gamma_2 \vdash J)$ donc que $\Psi(\mathbf{E}^*(d)) \sim \Psi(d)$.

□

Lemme 4.47. *Soit d une $UTT^r[R]$-dérivation et **E** l'algorithme vu au lemme 4.23. Si $\Psi(d)$ est définie alors $\Psi(\boldsymbol{E}(d))$ est également définie, de plus $\Psi(\boldsymbol{E}(d)) \sim \Psi(d)$.*

Démonstration. Par définition on sait que $\mathbf{E}(d) \equiv \textbf{less}(\mathbf{E}^*(d))$.
En appliquant les lemmes 4.46 et 4.45, on sait que $\Psi(\textbf{less}(\mathbf{E}^*(d)))$ est définie, de plus $\Psi(d) \sim \Psi(\mathbf{E}^*(d)) \sim \Psi(\textbf{less}(\mathbf{E}^*(d))) \equiv \Psi(\mathbf{E}(d))$.

□

Dans le théorème qui suit, nous considérons les algorithmes concernant les jugements présupposés dans leurs versions finales, c'est-à-dire celles qui s'appliquent directement à des $UTT^r[R]$-dérivations. Nous établissons la relation entre ces algorithmes et la transformation Ψ.

Theorème 4.48. *Soit d une $UTT^r[R]$-dérivation.*

- *Si d est une dérivation d'un des jugements :*
  $\Gamma, \Gamma' \vdash J$, $\Gamma, x : K, \Gamma' \vdash J$, $\Gamma \vdash K_1 = K_2$, $\Gamma \vdash k_1 = k_2 : K$ *ou* $\Gamma \vdash \Sigma : K$ *(Σ dénote un terme ou une égalité de termes), et si $\Psi(d)$ est définie alors respectivement :*

  (a) $\Psi(V_{\Gamma_1}(d))$ *est définie et* $\Psi_{V_{\Gamma_1}(d)}(\Gamma \vdash \textbf{\textit{valid}}) = \Psi_d(\Gamma \vdash \textbf{\textit{valid}})$ *;*

  (b) $\Psi(K_{\Gamma_1}(d))$ *est définie et* $\Psi_{K_{\Gamma_1}(d)}(\Gamma \vdash K\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K\textbf{\textit{kind}})$ *;*

  (c) $\Psi(\textbf{\textit{l}}_=(d))$ *et* $\Psi(\textbf{\textit{r}}_=(d))$ *sont définies, de plus* $\Psi_{\textbf{\textit{l}}_=(d)}(\Gamma \vdash K_1\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K_1\textbf{\textit{kind}})$ *et* $\Psi_{\textbf{\textit{r}}_=(d)}(\Gamma \vdash K_2\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K_2\textbf{\textit{kind}})$ *;*

  (d) $\Psi(l_=(d))$ *et* $\Psi(r_=(d))$ *sont définies, de plus* $\Psi_{l_=(d)}(\Gamma \vdash k_1 : K) = \Psi_d(\Gamma \vdash k_1 : K)$ *et* $\Psi_{r_=(d)}(\Gamma \vdash k_2 : K) = \Psi_d(\Gamma \vdash k_2 : K)$ *;*

  (e) $\Psi(\textbf{\textit{kd}}(d))$ *est définie et* $\Psi_{\textbf{\textit{kd}}(d)}(\Gamma \vdash K\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K\textbf{\textit{kind}})$.

- *Si d une dérivation du jugement* $\Gamma \vdash A <_c B : \textbf{\textit{Type}}$ *(ou* $\Gamma \vdash K <_c K'$*) et si $\Psi(d)$ est définie alors :*

  f) $\Psi(\textbf{\textit{l}}_<(d))$ *est définie et* $\Psi_{\textbf{\textit{l}}_<(d)}(\Gamma \vdash A : \textbf{\textit{Type}}) = \Psi_d(\Gamma \vdash A : \textbf{\textit{Type}})$
  (ou $\Psi_{\textbf{\textit{l}}_<(d)}(\Gamma \vdash K\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K\textbf{\textit{kind}})$*) ;*

  g) $\Psi(\textbf{\textit{r}}_<(d))$ *est définie et* $\Psi_{\textbf{\textit{r}}_<(d)}(\Gamma \vdash B : \textbf{\textit{Type}}) = \Psi_d(\Gamma \vdash B : \textbf{\textit{Type}}))$
  (ou $\Psi_{\textbf{\textit{r}}_<(d)}(\Gamma \vdash K'\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K'\textbf{\textit{kind}})$*) ;*

  h) $\Psi(\textbf{\textit{co}}(d))$ *est définie et* $\Psi_{\textbf{\textit{co}}(d)}(\Gamma \vdash c : (El(A))El(B)) = \Psi_d(\Gamma \vdash c : (El(A))El(B))$
  (ou $\Psi_{\textbf{\textit{co}}(d)}(\Gamma \vdash c : (K)K') = \Psi_d(\Gamma \vdash c : (K)K')$*).*

Démonstration. Prenons le cas de l'exemple f), la méthode est similaire pour les cas de c) à h).
La dérivation d est donc une dérivation du jugement $\Gamma \vdash A <_c B$: **Type**. En appliquant le lemme 4.46, on sait que $\Psi(\mathbf{E}^*(d))$ est définie et que $\Psi(\mathbf{E}^*(d)) \sim \Psi(d)$.
Ce qui implique que :
(1)$\Psi_d(\Gamma \vdash A : \textbf{Type}) = \Psi_{\mathbf{E}^*(d)}(\Gamma \vdash A : \textbf{Type})$.
De plus, nous savons que $\mathbf{E}^*(d)$ existe et est une $UTT^r[R]^-_w$-dérivation de $\Gamma \vdash A <_c B$: **Type**. Nous sommes en mesure d'appliquer le lemme 4.43 à $\mathbf{E}^*(d)$, ce qui nous permet d'affirmer que $\Psi(\mathbf{l}^\circ_<(\mathbf{E}^*(d)))$ est définie et que :
(2) $\Psi_{\mathbf{l}^\circ_<(\mathbf{E}^*(d))}(\Gamma \vdash A : \textbf{Type}) = \Psi_{\mathbf{E}^*(d)}(\Gamma \vdash A : \textbf{Type})$.
De plus, grâce au lemme 4.45, on sait que $\Psi(\textbf{less}(\mathbf{l}^\circ_<(\mathbf{E}^*(d))))$ existe et que :
$\Psi(\textbf{less}(\mathbf{l}^\circ_<(\mathbf{E}^*(d)))) \sim \Psi(\mathbf{l}^\circ_<(\mathbf{E}^*(d)))$.
Le jugement final de $\Psi(\mathbf{l}^\circ_<(\mathbf{E}^*(d)))$ étant $\Psi_{\mathbf{l}^\circ_<(\mathbf{E}^*(d))}(\Gamma \vdash A : \textbf{Type})$, on peut déduire l'égalité :
(3) $\Psi_{\mathbf{l}^\circ_<(\mathbf{E}^*(d))}(\Gamma \vdash A : \textbf{Type}) = \Psi_{\textbf{less}(\mathbf{l}^\circ_<(\mathbf{E}^*(d)))}(\Gamma \vdash A : \textbf{Type})$.
Or $\mathbf{l}_<(d) \equiv \textbf{less}(\mathbf{l}^\circ_<(\mathbf{E}^*(d)))$ donc on a :
(4) $\Psi_{\textbf{less}(\mathbf{l}^\circ_<(\mathbf{E}^*(d)))}(\Gamma \vdash A : \textbf{Type}) = \Psi_{\mathbf{l}_<(d)}(\Gamma \vdash A : \textbf{Type})$.
A partir des égalités (1), (2), (3) et (4) on peut conclure que
$\Psi_d(\Gamma \vdash A : \textbf{Type}) = \Psi_{\mathbf{l}_<(d)}(\Gamma \vdash A : \textbf{Type})$.

□

Lemme 4.49. *Soit d une $UTT^r[R]$-dérivation.*

- *Si d est une dérivation du jugement $\Gamma \vdash (x : K)Q$**kind** et si $\Psi(d)$ est définie alors :*

  (a) $\Psi(\boldsymbol{P}\_(d))$ *est définie et* $\Psi_{\boldsymbol{P}\_(d)}(\Gamma \vdash K\textbf{\textit{kind}}) = \Psi_d(\Gamma \vdash K\textbf{\textit{kind}})$ *;*

  (b) $\Psi(\boldsymbol{P}'\_(d))$ *est définie et* $\Psi_{\boldsymbol{P}'\_(d)}(\Gamma, x : K \vdash Q\textbf{\textit{kind}}) = \Psi_d(\Gamma, x : K \vdash Q\textbf{\textit{kind}})$.

- *Si d une dérivation du jugement $\Gamma \vdash (x : K)Q = (x : K')Q'$ et si $\Psi(d)$ est définie alors :*

  c) $\Psi(\boldsymbol{P}_=(d))$ *est définie et* $\Psi_{\boldsymbol{P}_=(d)}(\Gamma \vdash K = K') = \Psi_d(\Gamma \vdash K = K')$ *;*

d) $\Psi(\mathbf{P}'_{=}(d))$ *est définie et* $\Psi_{\mathbf{P}'_{=}(d)}(\Gamma, x : K \vdash Q = Q') = \Psi_d(\Gamma, x : K \vdash Q = Q')$ *;*

e) $\Psi(\mathbf{P}''_{=}(d))$ *est définie et* $\Psi_{\mathbf{P}''_{=}(d)}(\Gamma, x : K' \vdash Q = Q') = \Psi_d(\Gamma, x : K' \vdash Q = Q')$*.*

Démonstration.

a) Pour une $UTT^r[R]$-dérivation d, on a $\mathbf{P}_{\_}(d) \equiv K^{\circ}_{\Gamma_1}(p\_^{\circ}(\mathbf{E}(d)))$ et $\mathbf{P}'_{\_}(d) \equiv p\_^{\circ}(\mathbf{E}(d))$. On peut conclure à l'aide des lemmes 4.47 et 4.41, par une démonstration similaire à celle du théorème 4.48.

b) On procède par induction sur la taille de d.
Cas de base. Quand $\Psi(d) \equiv d$, évident.

Pas d'induction. Les règles à considérer sont 2.1, 2.2, 2.3, 5.2.

- Considérons le cas

$$d \equiv \frac{\Gamma, x : K \overset{d_1}{\vdash} Q = Q' \qquad \Gamma \overset{d_2}{\vdash} K = K'}{\Gamma \vdash (x : K)Q = (x : K')Q'}\ (5.2)$$

Nous avons $\mathbf{P}_{=}(d) \equiv \mathbf{E}(d_2)$, $\quad \mathbf{P}'_{=}(d) \equiv \mathbf{E}(d_1)$ $\quad$ et

$$\mathbf{P}''_{=}(d) \equiv \frac{\Gamma, x : K \overset{\mathbf{E}(d_1)}{\vdash} Q = Q' \qquad \Gamma \overset{\mathbf{E}(d_2)}{\vdash} K = K'}{\Gamma, x : K' \vdash Q = Q'}\ (3.3)$$

Grâce au lemme 4.47, on peut aisément conclure que $\Psi(\mathbf{P}_{=}(d))$ et $\Psi(\mathbf{P}'_{=}(d))$ sont définies et que :
$\Psi_{\mathbf{P}_{=}(d)}(\Gamma \vdash K = K') = \Psi_d(\Gamma \vdash K = K')$, $\quad$ et
$\Psi_{\mathbf{P}'_{=}(d)}(\Gamma, x : K \vdash Q = Q') = \Psi_d(\Gamma, x : K \vdash Q = Q')$.

Pour prouver que $\Psi(\mathbf{P}''_{=}(d))$ est définie, il suffit de prouver l'égalité :
$\Psi_{\mathbf{E}(d_1)}(\Gamma \vdash K\mathbf{kind}) = \Psi_{\mathbf{E}(d_2)}(\Gamma \vdash K\mathbf{kind})$.
Grâce au lemme 4.47, on peut déduire que :
$\Psi_{\mathbf{E}(d_1)}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_1}(\Gamma \vdash K\mathbf{kind})$ $\quad$ et $\quad$ $\Psi_{\mathbf{E}(d_2)}(\Gamma \vdash K\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash K\mathbf{kind})$.
De plus par définition de $\Psi(d)$, on a les égalités :

$\Psi_{d_1}(\Gamma \vdash K\mathbf{kind}) = \Psi_d(\Gamma \vdash K\mathbf{kind})$ et $\Psi_{d_2}(\Gamma \vdash K\mathbf{kind}) = \Psi_d(\Gamma \vdash K\mathbf{kind})$. A partir des égalités précédentes, on peut déduire l'égalité recherchée et conclure que $\Psi(\mathbf{P}''_{=}(d))$ est définie.
Il reste à prouver que :
$\Psi_{\mathbf{P}''_{=}(d)}(\Gamma, x : K' \vdash Q = Q') = \Psi_d(\Gamma, x : K' \vdash Q = Q')$.
Par définition de $\Psi(\mathbf{P}''_{=}(d))$ et grâce au lemme 4.47, on a :
$\Psi_{\mathbf{P}''_{=}(d)}(\Gamma, x : K' \vdash Q = Q') = \Psi_{\mathbf{E}(d_1)}(\Gamma), x : \Psi_{\mathbf{E}(d_2)}(K') \vdash \Psi_{\mathbf{E}(d_1)}(Q) = \Psi_{\mathbf{E}(d_1)}(Q')) =$
$\Psi_{d_1}(\Gamma), x : \Psi_{d_2}(K') \vdash \Psi_{d_1}(Q) = \Psi_{d_1}(Q'))$.
Or $\Psi_d(\Gamma, x : K' \vdash Q = Q') = \Psi_{d_1}(\Gamma), x : \Psi_{d_2}(K') \vdash \Psi_{d_1}(Q) = \Psi_{d_1}(Q'))$.
On peut donc conclure que :
$\Psi_{\mathbf{P}''_{=}(d)}(\Gamma, x : K' \vdash Q = Q') = \Psi_d(\Gamma, x : K' \vdash Q = Q')$.

$\square$

Lemme 4.50. *Soit une $UTT^r[R]$-dérivation d du jugement $\Gamma \vdash f(k) :$ **Type**.*

1. *Le terme $f(k)$ provient d'une application non coercitive.*
   Si $\Psi(d)$ est définie alors $\Psi(\boldsymbol{kl}_1(d))$ et $\Psi(\boldsymbol{kl}_2(d))$ sont définies et on a :
   $\Psi_d(\Gamma \vdash f : (K)\boldsymbol{Type}) = \Psi_{\boldsymbol{kl}_1(d)}(\Gamma \vdash f : (K)\boldsymbol{Type})$ *et*
   $\Psi_d(\Gamma \vdash k : K) = \Psi_{\boldsymbol{kl}_2(d)}(\Gamma \vdash k : K)$.

2. *Le terme $f(k)$ provient d'une application coercitive.*
   Si $\Psi(d)$ est définie alors $\Psi(\boldsymbol{kl}_1(d))$, $\Psi(\boldsymbol{kl}_2(d))$ et $\Psi(\boldsymbol{kl}_3(d))$ sont définies et on a :
   $\Psi_d(\Gamma \vdash f : (K)\boldsymbol{Type}) = \Psi_{\boldsymbol{kl}_1(d)}(\Gamma \vdash f : (K)\boldsymbol{Type})$,
   $\Psi_d(\Gamma \vdash k : K_0) = \Psi_{\boldsymbol{kl}_2(d)}(\Gamma \vdash k : K_0)$, *et*
   $\Psi_d(\Gamma \vdash K_0 <_c K) = \Psi_{\boldsymbol{kl}_3(d)}(\Gamma \vdash K_0 <_c K)$.

Démonstration. Preuve par induction sur les dérivations suivant le modèle des preuves des lemmes 4.41 et 4.43.

$\square$

Les lemmes suivants nous seront utiles pour démontrer que l'application Ψ est définie pour les dérivations faisant intervenir des types inductifs.

Lemme 4.51. *Considérons une petite sorte A. Notons $\Psi(A)$ la pré-sorte A dans laquelle des coercions ont été insérées relativement à une dérivation quelconque. $\Psi(A)$ est une petite sorte.*

Démonstration. Par induction sur la structure de A.
Cas de base. $A \equiv El(M)$, on a $\Psi(A) \equiv El(\Psi(M))$. $\Psi(A)$ est une petite sorte par définition.
Pas d'induction. $A \equiv (x : A_1)A_2$, où A_1 et A_2 sont des petites sortes. On a $\Psi(A) \equiv (x : \Psi(A_1))\Psi(A_2)$. Par hypothèse d'induction et par définition, $\Psi(A)$ est une petite sorte.

□

Lemme 4.52. *Considérons une sorte Φ telle que $PPOS_X(\Phi)$. Notons $\Psi(\Phi)$ la pré-sorte Φ dans laquelle des coercions ont été insérées relativement à une dérivation quelconque, on a $PPOS_X(\Psi(\Phi))$.*

Démonstration. Par induction sur la structure de Φ.
Cas de base. $\Phi \equiv X$, on a $\Psi(\Phi) \equiv X$, d'où par hypothèse, $PPOS_X(\Psi(\Phi))$.
Pas d'induction. $\Phi \equiv (x : A)\Phi_0$ où A est une petite sorte et $PPOS_X(\Phi_0)$. On a $\Psi(\Phi) \equiv (x : \Psi(A))\Psi(\Phi_0)$, par hypothèse d'induction on a $PPOS_X(\Psi(\Phi_0))$ et grâce au lemme 4.51 on sait que $\Psi(A)$ est une petite sorte. On peut conclure que $PPOS_X(\Psi(\Phi))$.

□

Lemme 4.53. *Considérons une sorte Θ telle que $PSCH_X(\Theta)$. Notons $\Psi(\Theta)$ la pré-sorte Θ dans laquelle des coercions ont été insérées relativement à une dérivation quelconque, on a $PSCH_X(\Psi(\Theta))$.*

Démonstration. Par induction sur la structure de Θ.
Cas de base. $\Theta \equiv X$, on a $\Psi(\Theta) \equiv X$. Par hypothèse on a $PSCH_X(\Psi(\Theta))$.
Pas d'induction.

- $\Theta \equiv (x : A)\Theta_0$ où A est une petite sorte et $PSCH_X(\Theta_0)$. On a $\Psi(\Theta) \equiv (x : \Psi(A))\Psi(\Theta_0)$, par hypothèse d'induction on a $PSCH_X(\Psi(\Theta_0))$ et grâce au lemme 4.51 on sait que $\Psi(A)$ est une petite sorte. On peut conclure que $PSCH_X(\Psi(\Theta))$.

- $\Theta \equiv (\Phi)\Theta_0$ où $PPOS_X(\Phi)$ et $PSCH_X(\Theta_0)$. On a $\Psi(\Theta) \equiv (\Psi(\Phi))\Psi(\Theta_0)$, par hypothèse d'induction on a $PSCH_X(\Psi(\Theta_0))$ et grâce au lemme 4.52 on sait que $PPOS_X(\Psi(\Phi))$. On peut conclure que $PSCH_X(\Psi(\Theta))$.

□

Lemme 4.54. *Soit d une $UTT^r[R]$-dérivation de $\Gamma \vdash \kappa^X[\bar{\Theta}] : K$. Si $\Psi(d)$ est définie alors les dérivations $\Psi(K_{\Theta_i}(d))$ sont définies, de plus*
$\Psi_d(\Gamma, X : \textbf{\textit{Type}} \vdash \Theta_i \textbf{\textit{kind}}) = \Psi_{K_{\Theta_i}(d)}(\Gamma, X : \textbf{\textit{Type}} \vdash \Theta_i \textbf{\textit{kind}})$ *pour* $1 \leq i \leq n$.

Démonstration. Preuve par induction sur les dérivations suivant le modèle des preuves des lemmes 4.41 et 4.43.

□

Lemme 4.55. *Les structures de $\Theta^\circ[A, C, z]$ et $\Phi^\natural[A, C, f, z]$ sont préservées après application de la transformation Ψ relative à une dérivation d, autrement dit*
$\Psi(\Theta^\circ[A, C, z]) = \Theta^\circ[\Psi(A), \Psi(C), \Psi(z)]$ *et*
$\Psi(\Phi^\natural[A, C, f, z]) = \Phi^\natural[\Psi(A), \Psi(C), \Psi(f), \Psi(z)]$. *Nous rappelons que*
$\Theta^\circ[A, C, z] =_{df}$
$(x_1 : M_1(A)) \ldots (x_n : M_n(A))(\Phi^\circ_{i_1}[A, C, x_{i_1}]) \ldots (\Phi^\circ_{i_k}[A, C, x_{i_k}])C(z(x_1, \ldots, x_n))$,
et que
$\Phi^\natural[A, C, f, z] =_{df} [x_1 : A_1] \ldots [x_n : A_n]f(z(x_1, \ldots, x_n))$.
Nous notons $\Psi(t)$ le terme t dans lequel des coercions ont été insérées relativement à une dérivation quelconque. Les égalités sont dans $UTT^r[R]$ par rapport à un contexte approprié.

Démonstration. Nous prouvons ici que la structure de $\Theta^\circ[A, C, z]$ est préservée après application de la transformation Ψ. Pour cela, il nous faut d'abord montrer que la structure de $\Phi^\circ[A, C, z]$ est préservée après application de la transformation Ψ. Rappelons que
$\Phi^\circ[A, C, z] =_{df} (x_1 : A_1) \ldots (x_n : A_n)C(z(x_1, \ldots, x_n))$. La structure est préservée s'il n'existe pas, dans $\Psi(\Phi^\circ[A, C, z])$, de coercion entre $\Psi(C)$ et $\Psi(z(x_1, \ldots, x_n))$, entre $\Psi(z)$ et $\Psi(x_1)$, ..., et entre $\Psi(z(x_1, \ldots, x_{n-1}))$ et $\Psi(x_n)$.
Vérifions qu'il n'existe pas de coercion entre $\Psi(C)$ et $\Psi(z(x_1, \ldots, x_n))$.
Nous savons par définition des opérateurs relatifs aux types inductifs que la sorte

de C est (A)**Type** et que la sorte de $z(x_1,\ldots,x_n)$ est A. Nous pouvons obtenir les jugements de typage de ces termes par rapport à deux dérivations d et d' et un contexte Γ. Nous avons $\Gamma \overset{d}{\vdash} C : (A)\mathbf{Type}$ et $\Gamma \overset{d'}{\vdash} z(x_1,\ldots,x_n) : A$. On suppose que l'application entre C et $z(x_1,\ldots,x_n)$ est coercitive. En appliquant le lemme 4.26 à la dérivation d, on a les dérivations $\mathbf{kl}_1(d)$, $\mathbf{kl}_2(d)$ et $\mathbf{kl}_3(d)$ des jugements $\Gamma \vdash C : (A)\mathbf{Type}$, $\quad \Gamma \vdash z(x_1,\ldots,x_n) : A_0 \quad$ et $\quad \Gamma \vdash A_0 <_c A$ respectivement. En appliquant le lemme 4.56 à $\mathbf{kl}_2(d)$ et d', on a $\Psi_{d'}(\Gamma \vdash z(x_1,\ldots,x_n) : A_0) = \Psi_{\mathbf{kl}_2(d)}(\Gamma \vdash z(x_1,\ldots,x_n) : A)$, et on peut en extraire à l'aide du lemme 4.50 le jugement
(1) $\Psi_d(\Gamma) \vdash \Psi_{\mathbf{kl}_2(d)}(A_0) = \Psi_{d'}(A)$.
De même que l'on peut obtenir grâce au lemme 4.50 l'égalité
$\Psi_{\mathbf{kl}_3(d)}(\Gamma \vdash A_0 <_c A) = \Psi_d(\Gamma \vdash A_0 <_c A)$, et en extraire le jugement
(2) $\Psi_d(\Gamma) \vdash \Psi_{\mathbf{kl}_2(d)}(A_0) <_{\Psi_{\mathbf{kl}_3(d)}(c)} \Psi_{d'}(A)$.
Les jugement (1) et (2) étant dans $UTT^r[R]_{ok}$, cela nous mène à une contradiction à cause du théorème 4.37. L'application entre C et $z(x_1,\ldots,x_n)$ n'est pas coercitive, il n'y a donc pas de coercion entre $\Psi(C)$ et $\Psi(z(x_1,\ldots,x_n))$.
Nous poursuivons le même raisonnement pour montrer qu'il n'existe pas de coercion entre $\Psi(z)$ et $\Psi(x_1)$, ..., et entre $\Psi(z(x_1,\ldots,x_{n-1}))$ et $\Psi(x_n)$.
Nous pouvons affirmer que la structure de $\Phi^\circ[A,C,z]$ est préservée après application de la transformation Ψ, autrement dit $\Psi(\Phi^\circ[A,C,z]) =_{df} \Phi^\circ[\Psi(A),\Psi(C),\Psi(z)]$.
En suivant un raisonnement analogue, nous prouvons que la structure de $\Theta^\circ[A,C,z]$ est préservée après application de la transformation Ψ.

□

4.2.4.3 Ψ est totale

Lemme 4.56. *Présumons que les conditions de cohérence soient satisfaites pour le sous-typage. Soit d et d′ deux $UTT^r[R]^-$-dérivations du même jugement : a) $\Gamma \vdash$ **valid** ou b) $\Gamma \vdash K$**kind** ; ou de deux jugements : c) $\Gamma \vdash k : K$ et $\Gamma \vdash k : K'$.*
Si $\Psi(d)$ et $\Psi(d')$ sont définies alors on a dans UTT^r :

a) $\Psi_d(\Gamma \vdash \textbf{\textit{valid}}) = \Psi_{d'}(\Gamma \vdash \textbf{\textit{valid}})$,

b) $\Psi_d(\Gamma \vdash K\textbf{\textit{kind}}) = \Psi_{d'}(\Gamma \vdash K\textbf{\textit{kind}})$,

c) $\Psi_d(\Gamma \vdash k : K) = \Psi_{d'}(\Gamma \vdash k : K')$.

Démonstration. Preuve par induction sur la somme des tailles des dérivations d et d'.
Cas de base. Quand d et d' sont des UTT^r-dérivations, évident.

Pas d'induction.

a) Les deux dérivations se terminent par la règle 1.2.

$$d \equiv \frac{\overset{d_0}{\Gamma \vdash K\textbf{kind}} \quad x \notin FV(\Gamma)}{\Gamma, x : K \vdash \textbf{valid}}\,(1.2) \qquad d' \equiv \frac{\overset{d'_0}{\Gamma \vdash K\textbf{kind}} \quad x \notin FV(\Gamma)}{\Gamma, x : K \vdash \textbf{valid}}\,(1.2)$$

Par hypothèse d et d' sont définies, d'où d_0 et d'_0 le sont également (lemme 4.40). Par hypothèse d'induction $\Psi_{d_0}(\Gamma \vdash K\textbf{kind}) = \Psi_{d'_0}(\Gamma \vdash K\textbf{kind})$. $\Psi_{d_0}(\Gamma \vdash K\textbf{kind})$ et $\Psi_{d'_0}(\Gamma \vdash K\textbf{kind})$ étant les jugements finaux de $\Psi(d_0)$ et $\Psi(d'_0)$ respectivement, alors $\Psi(d_0) \sim \Psi(d'_0)$. Par définition de $\Psi(d)$ et $\Psi(d')$, et toujours à l'aide du lemme 4.40 nous pouvons conclure que $\Psi(d) \sim \Psi(d')$ et donc $\Psi_d(\Gamma, x : K \vdash \textbf{valid}) = \Psi_{d'}(\Gamma, x : K \vdash \textbf{valid})$.

b) La dernière règle d'une UTT^r-dérivation d'un jugement de la forme $\Gamma \vdash K\textbf{kind}$ est déterminée par la structure de la sorte K. Si K est une sorte-produit, cette dernière règle sera la règle 5.1. Si K est une El-sorte, la dernière règle sera la règle 4.2.
Le lemme 4.33 nous permet d'affirmer que seuls les cas suivants sont possibles : 4.2/4.2 ou 5.1/5.1.

- cas 4.2/4.2

$$d \equiv \frac{\overset{d_0}{\Gamma \vdash A : \textbf{Type}}}{\Gamma \vdash El(A)\textbf{kind}}\,(4.2) \qquad d' \equiv \frac{\overset{d'_0}{\Gamma \vdash A : \textbf{Type}}}{\Gamma \vdash El(A)\textbf{kind}}\,(4.2)$$

Par hypothèse d et d' sont définies, d'où d_0 et d'_0 le sont également (lemme 4.40). Par hypothèse d'induction $\Psi_{d_0}(\Gamma \vdash A : \textbf{Type}) = \Psi_{d'_0}(\Gamma \vdash A : \textbf{Type})$. $\Psi_{d_0}(\Gamma \vdash A : \textbf{Type})$ et $\Psi_{d'_0}(\Gamma \vdash A : \textbf{Type})$ étant les jugements finaux de $\Psi(d_0)$ et $\Psi(d'_0)$ respectivement, alors $\Psi(d_0) \sim \Psi(d'_0)$. Par définition de $\Psi(d)$ et $\Psi(d')$, et toujours à l'aide du lemme 4.40 nous pouvons conclure que $\Psi(d) \sim \Psi(d')$ et donc $\Psi_d(\Gamma \vdash El(A)\textbf{kind}) = \Psi_{d'}(\Gamma \vdash El(A)\textbf{kind})$.

- cas 5.1/5.1

$$d \equiv \frac{\Gamma, x : K_1 \overset{d_0}{\vdash} K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2\textbf{kind}}\,(5.1) \qquad d' \equiv \frac{\Gamma, x : K_1 \overset{d_0'}{\vdash} K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2\textbf{kind}}\,(5.1)$$

Par une démonstration analogue à celle utilisée pour les cas précédents, nous démontrons que $\Psi(d) \sim \Psi(d')$ et donc $\Psi_d(\Gamma, x : K_1 \vdash K_2\textbf{kind}) = \Psi_{d'}(\Gamma, x : K_1 \vdash K_2\textbf{kind})$.

c) Les combinaisons suivantes sont à considérer : 1.3/1.3 ; 3.1/- ; 5.3/5.3 ; 5.5/5.5 ; CA.1/CA.1. Notons que la combinaison 5.5/CA.1 ne peut se produire si notre système est cohérent.
Pour illustration, nous réalisons la preuve des cas suivants :

- cas 1.3/1.3

$$d \equiv \frac{\Gamma, x : K, \Gamma' \overset{d_0}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K}\,(1.3) \qquad d' \equiv \frac{\Gamma, x : K, \Gamma' \overset{d_0'}{\vdash} \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K}\,(1.3)$$

Par une démonstration analogue à celle utilisée pour les cas précédents, nous démontrons que $\Psi(d) \sim \Psi(d')$ et donc $\Psi_d(\Gamma, x : K, \Gamma' \vdash x : K) = \Psi_{d'}(\Gamma, x : K, \Gamma' \vdash x : K)$.

- cas 3.1/-
$\Psi(d)$ étant définie, il en est de même pour $\Psi(d_1)$ et $\Psi(d_2)$. On applique l'hypothèse d'induction à d_1 et d', ce qui donne :
(I) $\Psi_{d_1}(\Gamma \vdash k : K'') = \Psi_{d'}(\Gamma \vdash k : K')$.
Nous devons prouver que l'égalité suivante est dérivable :
(II)$\Psi_{d_1}(\Gamma \vdash k : K'') = \Psi_d(\Gamma \vdash k : K)$.
Afin de le démontrer, il nous faut pour cela prouver les égalités suivantes :
(1) $\vdash \Psi_{d_1}(\Gamma) = \Psi_d(\Gamma)$,
(2) $\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(K'') = \Psi_d(K)$,
(3) $\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(k) = \Psi_d(k) : \Psi_{d_1}(K'')$.

Montrons (1) $\vdash \Psi_{d_1}(\Gamma) = \Psi_d(\Gamma)$. On obtient ce jugement par définition de $\Psi(d)$.
Montrons (2) $\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(K'') = \Psi_d(K)$.
Par définition de $\Psi(d)$, nous sommes en mesure de trouver des dérivations des égalités suivantes : $\Psi_{d_1}(\Gamma) \vdash \Psi_{d_2}(K) = \Psi_d(K)$, $\quad \Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(K'') =$

$\Psi_{d_2}(K'')$ et $\Psi_{d_1}(\Gamma) \vdash \Psi_{d_2}(K'') = \Psi_{d_2}(K)$.
Considérons $h_1 \equiv$

$$\frac{\Psi_{d_1}(\Gamma) \overset{h_2}{\vdash} \Psi_{d_1}(K'') = \Psi_{d_2}(K) \qquad \Psi_{d_1}(\Gamma) \vdash \Psi_{d_2}(K) = \Psi_d(K)}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(K'') = \Psi_d(K)} \text{(2.3)}$$

avec $h_2 \equiv$

$$\frac{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(K'') = \Psi_{d_2}(K'') \qquad \Psi_{d_1}(\Gamma) \vdash \Psi_{d_2}(K'') = \Psi_{d_2}(K)}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(K'') = \Psi_{d_2}(K)} \text{(2.3)}$$

Montrons (3) $\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(k) = \Psi_d(k) : \Psi_{d_1}(K'')$.
On sait, par définition de $\Psi(d)$ que $\Psi_d(\Gamma \vdash k : K) = \Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(k) : \Psi_{d_2}(K)$, ainsi que $\Psi_d(\Gamma \vdash k : K) = \Psi_{d_1}(\Gamma) \vdash \Psi_d(k) : \Psi_{d_2}(K)$.
On peut construire la dérivation $h_3 \equiv$

$$\frac{\dfrac{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(k) : \Psi_{d_2}(K)}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(k) = \Psi_{d_1}(k) : \Psi_{d_2}(K)} \text{(2.4)} \qquad \dfrac{\Psi_{d_1}(\Gamma) \vdash \Psi_d(k) : \Psi_{d_2}(K)}{\Psi_{d_1}(\Gamma) \vdash \Psi_d(k) = \Psi_d(k) : \Psi_{d_2}(K)} \text{(2.4)}}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_1}(k) = \Psi_d(k) : \Psi_{d_2}(K)} \text{(2.6)}$$

On a aussi $h_4 \equiv$

$$\frac{\Psi_{d_1}(\Gamma) \overset{h_3}{\vdash} \Psi_{d_1}(k) = \Psi_d(k) : \Psi_{d_2}(K) \qquad \dfrac{\Psi_{d_1}(\Gamma) \overset{h_2}{\vdash} \Psi_{d_1}(K'') = \Psi_{d_2}(K)}{\Psi_{d_1}(\Gamma) \vdash \Psi_{d_2}(K) = \Psi_{d_1}(K'')} \text{(2.2)}}{\Psi_{d_1}(\Gamma) \overset{h}{\vdash} \Psi_{d_1}(k) = \Psi_d(k) : \Psi_{d_1}(K'')} \text{(3.2)}$$

Sachant que :
(I) $\Psi_{d_1}(\Gamma \vdash k : K'') = \Psi_{d'}(\Gamma \vdash k : K')$ et
(II)$\Psi_{d_1}(\Gamma \vdash k : K'') = \Psi_d(\Gamma \vdash k : K)$, on peut conclure que $\Psi_d(\Gamma \vdash k : K) = \Psi_{d'}(\Gamma \vdash k : K')$.

$\square$

Theorème 4.57 (Ψ est totale)**.** *Si R est cohérent, alors la transformation Ψ est définie pour toute dérivation dans $UTT^r[R]$.*

Démonstration. Preuve par induction sur la taille des dérivations. Comme nous l'avons vu lors de la définition de la transformation Ψ, les UTT^r-égalités qui doivent être insérées sont contenues dans les égalités entre certains jugements présupposés des prémisses.

Cas de base.Pour toutes $UTT^r[R]_{ok}$-dérivation d, par définition $\Psi(d)$ est définie et $\Psi(d) = d$.

Pas d'induction. Pour la plupart des cas, les théorèmes 4.24 et 4.48 ainsi que le lemme 4.56 suffisent pour prouver le pas d'induction. Certains cas comme ceux des règles SK.4, ϑ-eq, E-eq requiert d'avantage d'efforts.

- Considérons le cas $d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} k = k' : K \qquad \Gamma \overset{d_2}{\vdash} k' = k'' : K}{\Gamma \vdash k = k'' : K}$ (2.6)

$$\Psi(d) \equiv \dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} k_1 = k_1' : K_1 \qquad \dfrac{\Gamma_1 \overset{?_1}{\vdash} k_1' = k_2' : K_1 \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2' = k_2'' : K_2 \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2' = k_2'' : K_2}\,(3.3) \qquad \Gamma_1 \overset{?_3}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2' = k_2'' : K_1}\,(3.2)}{\Gamma_1 \vdash k_1' = k_2'' : K_1}\,(2.6)}{\Gamma_1 \vdash k_1 = k_2'' : K_1}\,(2.6)$$

Notons que $\Psi_{d_1}(\Gamma \vdash k = k' : K) \equiv \Gamma_1 \vdash k_1 = k_1' : K_1$ et $\Psi_{d_2}(\Gamma \vdash k' = k'' : K) \equiv \Gamma_2 \vdash k_2' = k_2'' : K_2$. Il nous suffit de prouver l'égalité $\Psi_{d_1}(\Gamma \vdash k' : K) = \Psi_{d_2}(\Gamma \vdash k' : K)$ pour démontrer que les égalités dénotées par $?_1$, $?_2$ et $?_3$ sont dérivables dans UTT^r.
Par hypothèse d'induction, $\Psi(d_1)$ et $\Psi(d_2)$ sont définies. Grâce au théorème 4.24, nous savons que les dérivations $r_=(d_1)$ et $l_=(d_2)$ du même jugement $\Gamma \vdash k' : K$ sont définies. En appliquant le lemme 4.56 à $r_=(d_1)$ et $l_=(d_2)$, on a :
(1) $\Psi_{r_=(d_1)}(\Gamma \vdash k' : K) = \Psi_{l_=(d_2)}(\Gamma \vdash k' : K)$.
De plus le théorème 4.48 nous permet d'avoir les égalités :
(2) $\Psi_{r_=(d_1)}(\Gamma \vdash k' : K) = \Psi_{d_1}(\Gamma \vdash k' : K)$ et
(3) $\Psi_{l_=(d_2)}(\Gamma \vdash k' : K) = \Psi_{d_2}(\Gamma \vdash k' : K)$. On peut conclure à partir des égalités (1), (2) et (3) que $\Psi_{d_1}(\Gamma \vdash k' : K) = \Psi_{d_2}(\Gamma \vdash k' : K)$ ou encore $\Gamma_1 \vdash k_1' : K_1 = \Gamma_2 \vdash k_2' : K_2$. Cette égalité contient les trois égalités recherchées, à savoir $?_1$, $?_2$ et $?_3$.

- Considérons le cas $d \equiv \dfrac{PSCH_X(\bar{\Theta}) \qquad \Gamma, X : \textbf{Type} \overset{d_0}{\vdash} \Theta_i \textbf{kind}}{\Gamma \vdash M^X[\bar{\Theta}] : \textbf{Type}}$ (C-M) $1 \leq i \leq n$.

  Nous avons $\Psi(d) \equiv \dfrac{PSCH_X(\bar{\Theta_0}) \qquad \Gamma_0, X_0 : \textbf{Type} \overset{\Psi(d_0)}{\vdash} \Theta_{0i} \textbf{kind}}{\Gamma_0 \vdash M^X[\bar{\Theta_0}] : \textbf{Type}}$ (C-M) $1 \leq i \leq n$.

  Notons que $\Psi_{d_0}(\Gamma, X : \textbf{Type} \vdash \Theta_i \textbf{kind}) \equiv \Gamma_0, X_0 : \textbf{Type} \vdash \Theta_{0i} \textbf{kind}$ et $\Psi(\bar{\Theta}) \equiv \bar{\Theta}_0$.
  Il nous faut juste démontrer qu'on a $PSCH_X(\bar{\Theta_0})$, ce que nous permet le lemme 4.53.

- Considérons le cas

  $$d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} Q' <_{c'} Q \qquad \Gamma, x' : Q' \overset{d_2}{\vdash} [c'(x')/x]K <_{c''} K' \qquad \Gamma, x : Q \overset{d_3}{\vdash} K\textbf{kind}}{\Gamma \vdash (x : Q)K <_c (x' : Q')K'} \text{ (SK.4)}$$

  avec $c \equiv [f : (x : Q)K][x' : Q']c''(f(c'(x')))$.

  Considérons la dérivation $D_2 \equiv$

$$\dfrac{\dfrac{\Gamma_2, x' : Q_2' \overset{\Psi(d_2)}{\vdash} [c_2'(x')/x]K_2 <_{c_2''} K_2' \qquad \overset{?_1}{\vdash} \Gamma_2, x' : Q_2' = \Gamma_1, x' : Q_1'}{\Gamma_1, x' : Q_1' \vdash [c_2'(x')/x]K_2 <_{c_2''} K_2'} \text{ (3.3)} \qquad \Gamma_1, x' : Q_1' \overset{?_2}{\vdash} [c_2'(x')/x]K_2 = [c_1'(x')/x]K_3}{\Gamma_1, x' : Q_1' \vdash [c_1'(x')/x]K_3 <_{c_2''} K_2'} \text{ (SK.6)}$$

  ainsi que $D_3 \equiv$

  $$\dfrac{\Gamma_3, x : Q_3 \overset{\Psi(d_3)}{\vdash} K_3\textbf{kind} \qquad \overset{?_3}{\vdash} \Gamma_3, x : Q_3 = \Gamma_1, x : Q_1}{\Gamma_1, x : Q_1 \vdash K_3\textbf{kind}} \text{ (3.3)}$$

  Finalement on a $\Psi(d) \equiv$

$$\frac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} Q_1' <_{c_1'} Q_1 \qquad \Gamma_1, x' : Q_1' \overset{D_2}{\vdash} [c_1'(x')/x]K_3 <_{c_2''} K_2' \qquad \Gamma_1, x : Q_1 \overset{D_3}{\vdash} K_3\mathbf{kind}}{\Gamma_1 \vdash (x : Q_1)K_3 <_{c^*} (x' : Q_1')K_2'} \text{(SK.4)}$$

avec $c^* \equiv [f : (x : Q_1)K_3][x' : Q_1']c_2''(f(c_1'(x')))$.

Rappelons que :
$\Psi_{d_1}(\Gamma \vdash Q' <_{c'} Q) \equiv \Gamma_1 \vdash Q_1' <_{c_1'} Q_1$,
$\Psi_{d_2}(\Gamma, x' : Q' \vdash [c'(x')/x]K <_{c''} K') \equiv \Gamma_2, x' : Q_2' \vdash [c_2'(x')/x]K_2 <_{c_2''} K_2'$, et
$\Psi_{d_3}(\Gamma, x : Q \vdash K\mathbf{kind}) \equiv \Gamma_3, x : Q_3 \vdash K_3\mathbf{kind}$.

Les égalités $?_1$ et $?_3$ sont contenues dans
$\Psi_{d_1}(\Gamma \vdash Q'\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash Q'\mathbf{kind})$, $\Psi_{d_1}(\Gamma, x : Q \vdash \mathbf{valid}) = \Psi_{d_3}(\Gamma, x : Q \vdash \mathbf{valid})$. Nous prouvons ces égalités par une méthode similaire à celle employée pour le cas 2.6.

La difficulté apparaît pour trouver une dérivation correspondante à $?_2$.
Le théorème 4.24 nous permet de considérer la dérivation $\Gamma, x' : Q' \overset{h_2}{\vdash} [c'(x')/x]K\mathbf{kind}$, avec $h_2 \equiv \mathbf{l}_<(d_2)$.
Nous savons grâce au lemme 4.48 que $\Psi(h_2)$ est définie et que :
$\Psi_{h_2}(\Gamma, x' : Q' \vdash [c'(x')/x]K\mathbf{kind}) = \Psi_{d_2}(\Gamma, x' : Q' \vdash [c'(x')/x]K\mathbf{kind})$,
donc on a :
(1) $\Psi_{h_2}(\Gamma, x' : Q' \vdash [c'(x')/x]K\mathbf{kind}) = \Gamma_2, x' : Q_2' \vdash [c_2'(x')/x]K_2\mathbf{kind}$.

Considérons aussi la dérivation suivante :

$$\frac{\dfrac{\Gamma, x : Q \overset{d_3}{\vdash} K\mathbf{kind} \qquad \Gamma, x' : Q' \overset{h_2'}{\vdash} \mathbf{valid}}{\Gamma, x' : Q', x : Q \vdash K\mathbf{kind}}\text{(wkn)} \qquad \dfrac{\dfrac{\Gamma \overset{h_1}{\vdash} c' : (Q')Q \qquad \Gamma, x' : Q' \overset{h_2'}{\vdash} \mathbf{valid}}{\Gamma, x' : Q' \vdash c' : (Q')Q}\text{(wkn)} \qquad \dfrac{\Gamma, x' : Q' \overset{h_2'}{\vdash} \mathbf{valid}}{\Gamma, x' : Q' \vdash x' : Q'}\text{(1.3)}}{\Gamma, x' : Q' \vdash c'(x') : Q}\text{(5.5)}}{\Gamma, x' : Q' \vdash [c'(x')/x]K\mathbf{kind}}\text{(6.2)}$$

avec $h_1 \equiv \mathbf{co}(d_1)$, $h_2' \equiv V_{\Gamma_1}(d_2)$. Appelons h la dérivation obtenue par élimination de (wkn) et de (6.2) de la dérivation ci-dessus. A partir de cette dérivation, grâce au théorème 4.48 et au lemme 4.47 on peut montrer que $\Psi(h)$ est définie et que
(2) $\Psi_h(\Gamma, x' : Q' \vdash [c'(x')/x]K\mathbf{kind}) = \Gamma_1, x' : Q_1' \vdash [c_1'(x')/x]K_3\mathbf{kind}$,

et de l'autre côté en appliquant le lemme 4.56 à h_2 et h on a
(3) $\Psi_{h_2}(\Gamma, x' : Q' \vdash [c'(x')/x]K\textbf{kind}) = \Psi_h(\Gamma, x' : Q' \vdash [c'(x')/x]K\textbf{kind})$.
On obtient à partir des égalités (1), (2) et (3) l'égalité :
$\Gamma_1, x' : Q'_1 \vdash [c'_1(x')/x]K_3\textbf{kind}) = \Gamma_2, x' : Q'_2 \vdash [c'(x')/x]K_2\textbf{kind}$. Cette égalité suppose la dérivabilité du jugement $\Gamma_1, x' : Q'_1 \vdash [c'_1(x')/x]K_3 = [c'(x')/x]K_2$, et en lui appliquant la règle (2.2) on obtient une dérivation du jugement recherché à savoir :
$\Gamma_1, x' : Q'_1 \vdash [c'(x')/x]K_2 = [c'_1(x')/x]K_3$.

- Considérons le cas $d \equiv$

$$\frac{\Gamma \overset{d_1}{\vdash} E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma \overset{d_2}{\vdash} E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m \qquad \Gamma \overset{d_3}{\vdash} k : F_n}{\Gamma \vdash E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota^X_{\sigma(1)}[\bar{\Theta}_m], ..., \iota^X_{\sigma(n)}[\bar{\Theta}_m])(k)) = E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A} \ (\vartheta\text{-eq})$$

Comme indiqué lors de la définition de la transformation Ψ pour le cas de la règle ϑ-eq, il nous faut prouver les égalités suivantes :

$\Psi_{d_1}(\Gamma \vdash F_m\textbf{kind}) = \Psi_{d_2}(\Gamma \vdash F_m\textbf{kind})$, $\Psi_{d_1}(\Gamma \vdash \textbf{valid}) = \Psi_{d_3}(\Gamma \vdash \textbf{valid})$ et $\Psi_{d_2}(\Gamma \vdash F_n\textbf{kind}) = \Psi_{d_3}(\Gamma \vdash F_n\textbf{kind})$.

1. Démontrons l'égalité $\Psi_{d_1}(\Gamma \vdash F_m\textbf{kind}) = \Psi_{d_2}(\Gamma \vdash F_m\textbf{kind})$.

   Les algorithmes **kd** et **P_** nous permettent de dériver le jugement $\Gamma \overset{\textbf{P\_}(\textbf{kd}(d_1))}{\vdash} F_m\textbf{kind}$. Le théorème 4.48 et le lemme 4.49 nous permettent d'obtenir
   (1) $\Psi_{\textbf{P\_}(\textbf{kd}(d_1))}(\Gamma \vdash F_m\textbf{kind}) = \Psi_{d_1}(\Gamma \vdash F_m\textbf{kind})$.

   Les algorithmes **kd** et **P′_** nous permettent de dériver le jugement $\Gamma, x : F_n \overset{\textbf{P}'\textbf{\_}(\textbf{kd}(d_2))}{\vdash} F_m\textbf{kind}$. Le théorème 4.48 et le lemme 4.49 nous permettent d'obtenir
   (2) $\Psi_{\textbf{P}'\textbf{\_}(\textbf{kd}(d_2))}(\Gamma, x : F_n \vdash F_m\textbf{kind}) = \Psi_{d_2}(\Gamma, x : F_n \vdash F_m\textbf{kind})$.

   Nous rappelons que nous pouvons dériver le jugement $\Gamma \vdash F_m\textbf{kind}$ de la manière suivante :

$$D_1 \equiv \frac{\Gamma, x : F_n \overset{\mathbf{P}'\_(\mathbf{kd}(d_2))}{\vdash} F_m\mathbf{kind} \qquad \Gamma \overset{d_3}{\vdash} k : F_n}{\Gamma \vdash F_m\mathbf{kind}}\ (6.2) \qquad \text{car } x \notin FV(F_m).$$

On a donc l'égalité :
(3) $\Psi_{D_1}(\Gamma \vdash F_m\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash F_m\mathbf{kind})$.

De plus, en appliquant le lemme 4.56 à $\mathbf{P}\_(\mathbf{kd}(d_1))$ et D_1, on obtient :
(4) $\Psi_{\mathbf{P}\_(\mathbf{kd}(d_1))}(\Gamma \vdash F_m\mathbf{kind}) = \Psi_{D_1}(\Gamma \vdash F_m\mathbf{kind})$.
Des égalités (1), (2), (3), (4) nous obtenons : $\Psi_{d_1}(\Gamma \vdash F_m\mathbf{kind}) = \Psi_{d_2}(\Gamma \vdash F_m\mathbf{kind})$.

2. Par un raisonnement analogue, nous démontrons les égalités : $\Psi_{d_1}(\Gamma \vdash \mathbf{valid}) = \Psi_{d_3}(\Gamma \vdash \mathbf{valid})$ et $\Psi_{d_2}(\Gamma \vdash F_n\mathbf{kind}) = \Psi_{d_3}(\Gamma \vdash F_n\mathbf{kind})$.

- Considérons le cas $d \equiv$

$$\frac{\begin{array}{c} PSCH_X(\bar{\Theta}) \qquad \Gamma, X : \mathbf{Type} \overset{d_{1_i}}{\vdash} \Theta_i\mathbf{kind} \qquad \Gamma \overset{d_2}{\vdash} C : (M^X[\bar{\Theta}])\mathbf{Type} \\ \Gamma \overset{d_{3_i}}{\vdash} f_i : \Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]] \qquad \Gamma \overset{d_{4_j}}{\vdash} a_j : M_j(M^X[\bar{\Theta}]) \qquad 1 \leq j \leq m \quad 1 \leq i \leq n \end{array}}{\Gamma \vdash E^X[\bar{\Theta}](C, \bar{f}, \iota_i^X[\bar{\Theta}](\bar{a})) = f_i(\bar{a}, \Phi_1^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_1], \ldots, \Phi_k^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_k]) : C(\iota_i^X[\bar{\Theta}](\bar{a}))}\ (E)$$

Avec $\bar{\Theta}$ de la forme $< \Theta_1, \ldots, \Theta_n >$, $\bar{f}$ qui représente $f_1, \ldots, f_n$, $\bar{a}$ qui représente $a_1, \ldots, a_m$ et $ARITY_X(\Theta) \equiv \Phi_1, \ldots, \Phi_k$. Rappelons que Θ_i est de la forme $(x_1 : M_1) \ldots (x_m : M_m)X$

Afin de prouver que $\Psi(d)$ est définie, nous devons montrer que :

1. Les structures de $PSCH_X(\bar{\Theta})$, $\Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]]$ et $\Phi_1^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_1], \ldots, \Phi_k^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_k]$ demeurent inchangées après l'application de la transformation Ψ ;

2. $\Psi_{d_{1_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind}) = \Psi_{d_2}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind})$;

3. $\Psi_{d_{1_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind}) = \Psi_{d_{3_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind})$;

4. Pour toutes les occurrences de C dans $\Gamma \overset{d_{3_i}}{\vdash} f_i : \Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]]$, nous avons les égalités $\Psi_{d_2}(C) = \Psi_{d_{3_i}}(C)$;

5. $\Psi_{d_{1_i}}(\Gamma \vdash M_j(M^X[\bar{\Theta}])\mathbf{kind}) = \Psi_{d_{4_j}}(\Gamma \vdash M_j(M^X[\bar{\Theta}])\mathbf{kind})$.

Démontrons maintenant les points cités plus haut.

1. Les lemmes 4.53, 4.55 nous permettent de conclure que les structures de $PSCH_X(\bar{\Theta})$, $\Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]]$ et $\Phi_1^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_1], \ldots, \Phi_k^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_k]$ demeurent inchangées après l'application de la transformation Ψ.

2. Démontrons les égalités :
   $\Psi_{d_{1_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind}) = \Psi_{d_2}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind})$.
   A partir de la dérivation d_2, nous pouvons obtenir les dérivations $K_{\Theta_i}(El^-(\mathbf{P}\_(\mathbf{kd}(d_2))))$ des jugements $\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind})$.
   Le théorème 4.48 et le lemme 4.49 nous permettent d'obtenir
   (1) $\Psi_{K_{\Theta_i}(El^-(\mathbf{P}\_(\mathbf{kd}(d_2))))}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind}) = \Psi_{d_2}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind})$.

   De plus, en appliquant le lemme 4.56 à $K_{\Theta_i}(El^-(\mathbf{P}\_(\mathbf{kd}(d_2))))$ et d_{1_i}, on obtient
   (2) $\Psi_{K_{\Theta_i}(El^-(\mathbf{P}\_(\mathbf{kd}(d_2))))}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind}) = \Psi_{d_{1_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind})$.
   On peut donc déduire que pour $1 \leq i \leq n$
   $\Psi_{d_2}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind}) = \Psi_{d_{1_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind})$.

3. Démontrons maintenant les égalités :
   $\Psi_{d_{1_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind}) = \Psi_{d_{3_i}}(\Gamma, X : \mathbf{Type} \vdash \Theta_i\mathbf{kind})$,
   Ces égalités sont démontrées en employant une méthode quasi-similaire à la méthode employée pour démontrer les égalités précédentes.
4. Nous cherchons à démontrer ici que pour toutes occurrences de C dans $\Gamma \overset{d_{3_i}}{\vdash} f_i : \Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]]$, nous avons les égalités $\Psi_{d_2}(C) = \Psi_{d_{3_i}}(C)$.

   Rappel

– Nous rappelons tout d'abord que Θ_i est de la forme $(x_1 : M_1), \ldots, (x_m : M_m)X$, que $< \Phi_{i_1}, \ldots, \Phi_{i_k} >$ est la sous séquence de $< M_1, \ldots, M_m >$ des opérateurs strictement positifs et que Φ_{i_v} est de la forme $(y_{i_{v_1}} : P_{i_{v_1}}), \ldots, (y_{i_{v_p}} : P_{i_{v_p}})X$, $(1 \leq v \leq k)$.

– Nous rappelons également que la sorte de f_i, à savoir $\Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]]$ est de la forme $(x_1 : M_1(M^X[\bar{\Theta}])) \ldots (x_n : M_n(M^X[\bar{\Theta}]))$ $(\Phi_{i_1}^\circ[M^X[\bar{\Theta}], C, x_{i_1}]) \ldots (\Phi_{i_k}^\circ[M^X[\bar{\Theta}], C, x_{i_k}])$ $C(\iota_i^X[\bar{\Theta}](x_1, \ldots, x_n))$; avec $\Phi_{i_v}^\circ[M^X[\bar{\Theta}], C, x_{i_v}] =_{df} (y_{i_{v_1}} : P_{i_{v_1}}) \ldots (y_{i_{v_p}} : P_{i_{v_p}})C(x_{i_v}(y_{i_{v_1}}, \ldots, y_{i_{v_p}}))$.

Commençons par la première occurrence de C.

Pour cela cherchons une dérivation de :
$\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash x_{i_1}(y_{i_{1_1}}, \ldots, y_{i_{1_p}}) : M^X[< \Theta_1, \ldots, \Theta_n >]$.

A partir des dérivations d_{1_i} et de $PSCH_X(\bar{\Theta})$, on peut dériver, en appliquant les algorithmes adéquats :
$\Gamma \vdash \Theta_i(M^X[\bar{\Theta}])$**kind**. Autrement dit
$\Gamma \vdash (x_1 : M_1), \ldots, (x_m : M_m)(M^X[\bar{\Theta}])$**kind**.
On peut ensuite obtenir $\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]) \vdash x_{i_1} : \Phi_{i_1}(M^X[\bar{\Theta}])$**kind**, ce qui équivaut à
$\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]) \vdash x_{i_1} : (y_{i_{1_1}} : P_{i_{1_1}}), \ldots, (y_{i_{1_p}} : P_{i_{1_p}})(M^X[\bar{\Theta}])$.

De même, on peut également construire les dérivations :
$\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash y_{i_{1_w}} : P_{i_{1_w}} \quad (1 \leq w \leq p)$.

On peut donc construire à partir des dérivations d_{1_i} une dérivation D_2 du jugement :
$\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash x_{i_1}(y_{i_{1_1}}, \ldots, y_{i_{1_p}}) : M^X[< \Theta_1, \ldots, \Theta_n >]$.

Cherchons maintenant une dérivation D_1 du jugement :
$\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} :$

$P_{i_{1_p}} \vdash C(x_{i_1}(y_{i_{1_1}}, \ldots, y_{i_{1_p}}))$: **Type**.

Pour chaque dérivation d_{3_i}, en utilisant les algorithmes concernant les jugements présupposés appropriés, nous sommes en mesure de construire la dérivation D_1.
En conséquence du lemme 4.55 qui stipule que la structure de $\Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]]$ est préservée après application de Ψ, nous pouvons affirmer que l'application entre C et $x_{i_1}(y_{i_{1_1}}, \ldots, y_{i_{1_p}})$ n'est pas coercitive. Connaissant également la dérivation D_2, nous sommes donc en mesure d'appliquer le lemme 4.26 et avoir une dérivation $\mathbf{kl}_1(D_1)$ du jugement
$\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash C : (M^X[\bar{\Theta}])$**Type**.
En appliquant le théorème 4.48, le lemme 4.49, le lemme 4.50, ainsi que le lemme 4.56 à $\mathbf{kl}_2(D_1)$ et D_2 on obtient les égalités suivantes :
(1) $\Psi_{\mathbf{kl}_1(D_1)}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash C : (M^X[\bar{\Theta}])\mathbf{Type}) =$
$\Psi_{D_1}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}}) \vdash \Psi_{D_1}(C) : (\Psi_{D_2}(M^X[\bar{\Theta}]))\mathbf{Type}$.

A l'aide de la règle (wkn), on construit à partir de d_2, la dérivation d_2' de
$\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash C : (M^X[\bar{\Theta}])$**Type**. Par conséquent on a :
(2) $\Psi_{d_2}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash C : (M^X[\bar{\Theta}])\mathbf{Type}) =$
$\Psi_{d_2'}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash C : (M^X[\bar{\Theta}])\mathbf{Type})$.

En appliquant le lemme 4.56 à $\mathbf{kl}_1(D_1)$ et d_2' on obtient :
(3) $\Psi_{\mathbf{kl}_1(D_1)}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash C : (M^X[\bar{\Theta}])\mathbf{Type}) =$
$\Psi_{d_2'}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}} \vdash C : (M^X[\bar{\Theta}])\mathbf{Type})$.
Après analyse des dérivations D_1 et D_2 nous pouvons affirmer que :
(4) $\Psi_{D_1}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} : P_{i_{1_p}}) =$

$\Psi_{d_{3_i}}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} :$ $P_{i_{1_p}})$, que
(5) $\Psi_{D_1}(C) = \Psi_{d_{3_i}}(C)$, et que
(6) $\Psi_{D_2}(M^X[< \Theta_1, \ldots, \Theta_n >]) =$
$M^X[< \Theta'_1, \ldots, \Theta'_n >]$, avec pour $1 \leq i \leq n$, $\Theta'_i \equiv \Psi_{d_{1_i}}(\Theta_i)$.

Nous pouvons conclure, à partir des égalités (1), (2), (3), (4), (5) et (6), l'égalité :
$\Psi_{d_2}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} :$ $P_{i_{1_p}} \vdash C : (M^X[\bar{\Theta}])\mathbf{Type}) =$
$\Psi_{d_{3_i}}(\Gamma, x_1 : M_1(M^X[\bar{\Theta}]), \ldots, x_m : M_m(M^X[\bar{\Theta}]), y_{i_{1_1}} : P_{i_{1_1}}, \ldots, y_{i_{1_p}} :$ $P_{i_{1_p}} \vdash \Psi_{d_{3_i}}(C) : (M^X[< \Theta'_1, \ldots, \Theta'_n >])\mathbf{Type}$, d'où
$\Psi_{d_2}(C) = \Psi_{d_{3_i}}(C)$.

Nous procédons de la même manière pour les autres occurrences des C.

5. Montrons maintenant que $\Psi_{d_{1_i}}(\Gamma \vdash M_j(M^X[\bar{\Theta}])\mathbf{kind}) = \Psi_{d_{4_j}}(\Gamma \vdash M_j(M^X[\bar{\Theta}])\mathbf{kind})$.

   A partir de la dérivation d_{1_i}, à l'aide des algorithmes sur les jugements présupposés et des règles de substitution, nous sommes en mesure de dériver les jugements
   $\Gamma \vdash M_j(M^X[\bar{\Theta}])\mathbf{kind}$, pour $1 \leq j \leq m$.
   En appliquant l'algorithme **kd** à d_{4_j}, on obtient $\Gamma \overset{\mathbf{kd}(d_{4_j})}{\vdash} M_j(M^X[\bar{\Theta}])\mathbf{kind}$.
   Les théorèmes 4.48 et 4.56, ainsi que le lemme 4.49, nous permettent de conclure.

$\square$

4.2.5 Conservativité

Theorème 4.58 (Conservativité)**.** *Soit J un jugement dérivable dans $UTT^r[R]$. Si R est cohérent et si J n'est pas un jugement de sous-typage ou de sous-sorte, alors*

J est dérivable dans UTT^r si et seulement si il existe une $UTT^r[R]$-dérivation d telle que $\Psi_d(J) \equiv J$.
Ce théorème reste valable si à la place de UTT^r on considère $UTT^r[R]_o$ ou $UTT^r[R]_{ok}$ et J étant respectivement un jugement de sous-typage ou de sous-sorte.

Démonstration. L'implication de la gauche vers la droite est triviale. En effet si J est dérivable dans UTT^r, alors il existe une UTT^r-dérivation de J qui est forcément une $UTT^r[R]$-dérivation. Cette dérivation ne contient donc pas de règle d'application coercitive et par définition de Ψ on a $\Psi_d(J) \equiv J$.
Voyons maintenant l'implication de la droite vers la gauche. Supposons qu'il existe une $UTT^r[R]$-dérivation d de J telle que $\Psi_d(J) \equiv J$. J n'étant pas un jugement de sous-typage ou de sous-sorte, on peut facilement constater que $\Psi(d)$ est une dérivation de J dans UTT^r. (Notons que $\Psi_d(J)$ est le jugement final de $\Psi(d)$).
Si J est un jugement de sous-typage ou de sous-sorte, alors $\Psi(d)$ est respectivement une $UTT^r[R]_o$-dérivation de J ou une $UTT^r[R]_{ok}$-dérivation de J.

□

Remarque 4.59. *Le théorème 4.58 nous permet de conclure sur la conservativité de UTT^r par rapport à $UTT^r[R]$. Nous rappelons que $UTT^r[R]_o$ est une extension conservative de UTT^r et que $UTT^r[R]_{ok}$ est une extension conservative de $UTT^r[R]_o$.*
Si le jugement J est dérivable dans $UTT^r[R]$ et ne contient pas d'application coercitive, on prend une de ses dérivations et on lui applique Ψ. Quelque soit la dérivation d du jugement J, on a $\Psi_d(J) \equiv J$ (théorème 4.58) et on sait que $\Psi(d)$ est dans $UTT^r[R]_o$, $UTT^r[R]_{ok}$ ou UTT^r. On peut conclure que le jugement J est dérivable respectivement dans $UTT^r[R]_o$, $UTT^r[R]_{ok}$ ou UTT^r selon que J soit un jugement de sous-typage, un jugement de sous-sorte ou aucun des deux. L'implication dans l'autre sens est évidente.

Conclusion

4.3 Synthèse

Pour rappel, cette thèse se situe dans le cadre de l'étude d'un système aux types dépendants, lequel est enrichi par la notion de sous-typage coercitif.

Les travaux de cette thèse s'inscrivent dans la continuité des travaux menés dans [SC03], dans lesquels la possibilité d'ajout de réductions non standards au sein du λ-calcul simplement typé, est étudiée. Dans notre cas, nous nous intéressons à un système intégrant entre autre des types dépendants, des types inductifs, un univers imprédicatif des propositions ainsi qu'un univers prédicatif, à savoir le système UTT. Dans un premier temps, nous avons ajouté au système UTT une réduction concernant une égalité extensionnelle pour les fonctions entre types finis, la ϑ-réduction. Nous avons nommé ce nouveau système UTT^r. Ajouter des réductions à une théorie de types, comporte les risques que les propriétés de normalisation forte, de préservation du type et de confluence ne soient plus vérifiées. C'est pour cela que nous avons suivi et adapté la méthode développée dans [Gog94] afin de garantir que les propriétés en question sont toujours préservées pour le système UTT^r. Un des intérets de UTT^r par rapport à la ϑ-réduction, est qu'il va grandement améliorer l'efficacité des calculs pour un fragment du système, celui des types finis.

Dans un second temps, nous nous sommes intéressés à l'enrichissement de UTT^r par la notion de sous-typage coercitif, à savoir le système $UTT^r[R]$. Le sous-typage coercitif étant vu comme un mécanisme d'abréviation, il a été primordial de garantir la conservativité de $UTT^r[R]$ par rapport UTT^r. Nous avons donc réalisé la preuve de conservativité de $UTT^r[R]$ par rapport UTT^r, preuve qui présente un double intérêt. En effet, UTT^r n'est autre que UTT auquel la règle d'égalité ϑ-eq a été rajoutée. De ce fait, la preuve de conservativité de $UTT^r[R]$ par rapport UTT^r, nous fournit également une preuve la conservativité de $UTT[R]$ par rapport UTT. Un autre effet de l'ajout de la ϑ-réduction, est d'augmenter l'ensemble des systèmes de coercions cohérents. Voyons l'exemple qui suit.

Exemple 4.60. *Prenons les types finis F_2, F_3, F_4, et F_5 de même que les coercions :*

- $c_1 \equiv E[\bar{\Theta}_3]([y : F_3]F_5, 1_{F_5}, 2_{F_5}, 3_{F_5}) : (F_3)F_5,$ *avec* $(\forall i_{F_3} : F_3, c_1(i_{F_3}) = i_{F_5})$
- $c_2 \equiv E[\bar{\Theta}_2]([z : F_2]F_3, 1_{F_3}, 2_{F_3}) : (F_2)F_3,$ *avec* $(\forall i_{F_2} : F_2, c_2(i_{F_2}) = i_{F_3})$
- $c_3 \equiv E[\bar{\Theta}_4]([y : F_4]F_5, 1_{F_5}, 2_{F_5}, 3_{F_5}, 4_{F_5}) : (F_4)F_5,$ *avec* $(\forall i_{F_4} : F_4, c_3(i_{F_4}) = i_{F_5})$ *et*
- $c_4 \equiv E[\bar{\Theta}_2]([z : F_2]F_4, 1_{F_4}, 2_{F_4}) : (F_2)F_4,$ *avec* $(\forall i_{F_2} : F_2, c_4(i_{F_2}) = i_{F_4})$

Le diagramme suivant commute.

$$\begin{tikzcd} F_2 \arrow[r, "c_2"] \arrow[d, "c_4"'] & F_3 \arrow[d, "c_1"] \\ F_4 \arrow[r, "c_3"'] & F_5 \end{tikzcd}$$

En effet, nous avons :
$c_1 \circ c_2(x) \rhd_\vartheta h(x)$ et $c_3 \circ c_4(x) \rhd_\vartheta h(x)$, avec
$h \equiv E[\bar{\Theta}_2]([x : F_2]F_5, 1_{F_5}, 2_{F_5}) : (F_2)F_5$.
Ce système est cohérent dans $UTT^r[R]$, car grâce à la ϑ-réduction, nous avons $c_1 \circ c_2(x) = c_3 \circ c_4(x)$.

4.4 Perspectives

Comme nous l'avons déjà précisé, l'intérêt d'un système comme $UTT^r[R]$ est qu'il permettra aux assistants à la preuve de gagner grandement en efficacité pour un fragment du calcul qui concerne les types finis. Ce fait est dû à l'ajout de la ϑ-réduction qui, non seulement facilitera les calcul pour les fonctions entre types finis mais autorisera également davantage de systèmes de coercions cohérents. La suite de ce travail se situe naturellement dans le développement d'implémentations. De plus, les perspectives de recherches ultérieures dans un cadre théorique sont nombreuses. Il serait tout d'abord intéressant de poursuivre l'étude de la possibilité d'ajout des autres réductions étudiées dans [SC03], dans un système aux types dépendants tel que le notre. Notre attention se porte déjà sur la réduction concernant les copies. Il serait également très utile de pouvoir explorer la question de la conservativité de manière générale pour toutes les théories déclarées dans LF. Cela passerait évidemment par la définition formelle de la sous-classe des théories pouvant être déclarées dans LF, pour lesquelles la conservativité serait assurée.

Annexe A

Résumé de la preuve de Healfdene Goguen

A.1 Le système UTT^S

Dans son chapitre 4, Goguen propose une sémantique opérationnelle pour le système UTT afin d'étudier les propriétés de ce dernier. Les réductions non-typées fournissent une sémantique opérationnelle naturelle pour les théories de types. Cependant, les réductions non-typées ne prennent pas en compte les informations sur le typage malgré l'importance de la relation entre les égalités calculatoires et le typage. De plus, elles ne donnent pas d'informations sur les objets canoniques c'est-à-dire les objets en forme normale incluant les constantes et les variables. Il introduit la sémantique opérationnelle typée qui définit une réduction vers la forme normale pour les termes qui sont bien typés dans la théorie des types. Il étudie une sémantique opérationnelle spécifique liée à la notion de réduction pour le calcul UTT.

En premier lieu, Goguen introduit plusieurs notions sur les termes qui seront importantes dans la présentation de UTT^S.

Définition A.1. *Soit R une relation de récriture sur les termes, la réduction $\triangleright_R$ est sa fermeture contextuelle.*

Définition A.2 (Réductions non typées)**.** *On introduit les relations de récriture suivantes :*

- $([x : A_1]M_0)(M_2) \quad \beta \quad [M_2/x]M_0$;
- $[x : A_1]M(x) \quad \eta \quad M \qquad x \notin FV(M)$;
- $E_\forall(A_1, P_1, R, f, \Lambda(A_2, P_2, g)) \quad o \quad f(g)$;
- $E^X[\bar{\Theta}_1](C, \bar{f}, \iota_i^X[\bar{\Theta}_2](\bar{a})) \quad o \quad f_i(\bar{a}, \Phi_{i_1}^{\natural}[M^X[\bar{\Theta}_3], C, E^X[\bar{\Theta}_3](C', \bar{f}'), a_i], \dots)$;
- $T(prop) \quad o \quad Prop$;
- $T(prf(P)) \quad o \quad Prf(P)$;
- $T(\mu^X[\bar{\Theta}]) \quad o \quad M^X[\bar{\Theta}]$.

On note $\triangleright$, pour désigner la fermeture contextuelle de toutes les récritures précédentes. Un terme M est un rédex s'il existe N tel que $M\beta\eta oN$.

Définition A.3. *On note $M \rhd_R^+ N$, pour la fermeture transitive de la réduction $\rhd_R$ et $M \rhd_R^* N$, pour la fermeture réflexive transitive de $\rhd_R$.*

Définition A.4. *Un terme est fortement normalisant, si toute séquence de réduction partant de ce terme termine.*

Définition A.5 (Pre-redex)**.** *On dit qu'un terme M est un full pre-redex, si M est une abstraction ou de la forme $E_\forall(A, P, R, f)$, $E^X[\bar{\Theta}](C, \bar{f})$ ou T.*

Si M est un full pre-redex et que M n'est pas une abstraction, alors on dit que M est un object-level full pre-redex.

Un terme qui est un full pre-redex ou un sous-terme de $E_\forall(A, P, R, f)$ ou $E^X[\bar{\Theta}](C, \bar{f})$, est un pre-redex.

Définition A.6 (terme de base)**.** *On définit la notion de terme de base inductivement de la manière suivante :*

- *les variables sont des termes de base ;*
- *si M_1 est un terme de base, alors $M_1(M_2)$ est un terme de base ;*
- *si M_2 est un terme de base et M_1 est un object-level full pre-redex, alors $M_1(M_2)$ est un terme de base.*

Définition A.7 (Forme normale de tête faible)**.** *Un terme M est en forme normale de tête faible, si M est un terme de base ou un pre-redex.*

Définition A.8 (Forme normale)**.** *Un terme M ou une sorte A est en forme normale si :*

- *M est une variable ;*
- *$A \equiv (x : A_1)A_2$ où A_1 et A_2 sont en forme normale ;*
- *$M \equiv [x : A_1]M_0$ où A_1 est en forme normale et M_0 est normal et pas de la forme $N(x)$ avec $x \notin FV(N)$;*

- *$M \equiv M_1(M_2)$ où M_1 et M_2 sont en forme normale et $M_1(M_2)$ n'est pas un redex ;*

- *$A \equiv$ **Type** ;*

- *$A \equiv El(M)$ et M est en forme normale ;*

- *M est une constante ;*

- *si $M \equiv \kappa^X[\bar{\Theta}]$, où Θ_i est en forme normale pour $1 \leq i \leq n$.*

A cause de la présence des règles de substitution qui sont admissibles, le système UTT n'est pas le plus pratique pour raisonner. Goguen introduit donc le calcul UTT^- dont les jugements sont les mêmes que ceux de UTT, mais qui ne contient pas de règles de substitution. Le symbole $\vdash^-$ est employé pour indiquer qu'un jugement a été dérivé dans UTT^-.

Goguen commence donc par prouver la complétude de UTT par rapport à UTT^-.

Lemme A.9 (Complétude de UTT par rapport à UTT^-)**.** *Si le jugement $\Gamma \vdash^- J$ est dérivable, alors $\Gamma \vdash J$ est dérivable.*

Goguen poursuit en définissant les formes de jugement pour UTT ainsi que leurs significations intuitives qui sont les suivantes :

- $\Gamma \vdash^S$ **valid** signifie Γ est un contexte valide ;

- $\Gamma \vdash^S A \xrightarrow{nf} B$, signifie que la sorte A admet une forme normale qui est la sorte B et que A et B sont des sortes valides dans le contexte Γ ;

- $\Gamma \vdash^S M \xrightarrow{nf} P : A$ signifie que le terme M admet une forme normale P de sorte A dans le contexte Γ ;

- $\Gamma \vdash^S M \xrightarrow{wh} N : A$ signifie que M se réduit en N en un pas par une stratégie externe de réduction, de plus M et N sont des termes de sorte A dans le contexte Γ.

Nous considérons également les abréviations suivantes :

- $\Gamma \vdash^S M : B$ s'il existe un terme P tel que $\Gamma \vdash^S M \xrightarrow{nf} P : B$;
- $\Gamma \vdash^S A$**kind**, s'il existe une sorte B telle que $\Gamma \vdash^S A \xrightarrow{nf} B$;
- $\Gamma \vdash^S M \downarrow N[P] : B$, si $\Gamma \vdash^S M \xrightarrow{nf} P : B$ et $\Gamma \vdash^S N \xrightarrow{nf} P : B$;
- $\Gamma \vdash^S A \downarrow B[C]$, si $\Gamma \vdash^S A \xrightarrow{nf} C$ et $\Gamma \vdash^S B \xrightarrow{nf} C$;
- $SCH^S_{\Gamma,X}(\bar{\Theta})$ si $\Gamma \vdash^S$ **valid**, $PSCH_X(\bar{\Theta})$ et Γ, X : **Type** $\vdash^S \Theta_i \xrightarrow{nf} \Theta'_i$ pour $1 \leq i \leq n$;
- $\Gamma \vdash^S \bar{\Theta} \downarrow \bar{\Theta}'[\bar{\Theta}'']$, si $\Gamma \vdash^S$ **valid**, $PSCH_X(\bar{\Theta})$, $PSCH_X(\bar{\Theta}')$ et Γ, X : **Type** $\vdash^S \Theta_i \downarrow \Theta'_i[\bar{\Theta}'']$ pour $1 \leq i \leq n$.

Puis, il propose quelques définitions et des lemmes concernant les propriétés de UTT^S. Voici les plus significatifs :

Lemme A.10 (Génération)**.** *Si un jugement est dérivable dans UTT^{rS}, alors quelque soit la dérivation de ce jugement elle se termine par une règle d'inférence unique et déterminée qui dépend de la forme de ce jugement.*

Lemme A.11 (Complétude pour UTT^-)**.**

- *Si* $\Gamma \vdash^S$ ***valid*** *alors* $\Gamma \vdash^-$ ***valid***.
- *Si* $\Gamma \vdash^S A \xrightarrow{nf} B$ *alors* $\Gamma \vdash^- A$***kind*** *et* $\Gamma \vdash^- A = B$.
- $\Gamma \vdash^S M \xrightarrow{nf} P : A$ *alors* $\Gamma \vdash^- M : A$ *et* $\Gamma \vdash^- M = P : A$.
- $\Gamma \vdash^S M \xrightarrow{wh} N : A$ *alors* $\Gamma \vdash^- M : A$ *et* $\Gamma \vdash^- M = N : A$.

Lemme A.12 (Unicité de la forme normale)**.**

- *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : B$ *et* $\Gamma \vdash^S M \xrightarrow{nf} Q : C$ *alors* $P \equiv Q$ *et* $B \equiv C$.
- *Si* $\Gamma \vdash^S A \xrightarrow{nf} B$ *et* $\Gamma \vdash^S A \xrightarrow{nf} C$ *alors* $B \equiv C$.

Définition A.13. *(Renommage). Un renommage est une substitution de* Δ *vers* Γ *telle que pour chaque* $(x : A) \in \Gamma$ *nous avons* $\delta(x) \equiv y$ *et* $(y : \hat{\delta}(A)) \in \Delta$.
Notation *:*$\hat{\delta}(M)$ *représente* M *dans lequel toutes les variables libres* v *sont remplacées par* $\delta(v)$.

Lemme A.14 (Renommage)**.** *Si* δ *est un renommage de* Δ *vers* Γ *alors :*

- *si* $\Gamma \vdash^S A \xrightarrow{nf} B$ *alors* $\Delta \vdash^S \hat{\delta}(A) \xrightarrow{nf} \hat{\delta}(B)$ *;*
- *si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$ *alors* $\Delta \vdash^S \hat{\delta}(M) \xrightarrow{nf} \hat{\delta}(P) : \hat{\delta}(A)$.

Le calcul UTT^S étant une synthèse des réductions standards et des règles de typage de UTT, il est en mesure de prouver des résultats concernant la relation entre les réductions non typées et le typage dans UTT^S.

Lemme A.15 (Adéquation pour réductions non typées)**.**

- *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$ *alors* P *est normal et il existe un* N *tel que* $M \rhd^*_{\beta o} N \rhd^*_{\eta} P$.
- *Si* $\Gamma \vdash^S A \xrightarrow{nf} C$ *alors* C *est normal et il existe un* B *tel que* $A \rhd^*_{\beta o} B \rhd^*_{\eta} C$.
- *Si* $\Gamma \vdash^S M \xrightarrow{wh} N : A$ *alors* $M \rhd_{\beta o} N$.

Pour finir, Goguen définit deux prédicats $S^C_\Gamma(A)$ et $S^{P,A}_\Gamma(M)$, dont la définition inclut les propriétés de normalisation forte, de préservation du type, et de confluence.

Définition A.16. *Les prédicats* $S^C_\Gamma(A)$ *et* $S^{P,A}_\Gamma(M)$ *sont définis comme la plus petite relation telle que :*

- $S^C_\Gamma(A)$ *si* $\Gamma \vdash^S A \xrightarrow{nf} C$ *et pour tout* B *si* $A \rhd B$ *alors* $S^C_\Gamma(B)$.
- $S^{P,A}_\Gamma(M)$ *si*
  - $\Gamma \vdash^S M \xrightarrow{nf} P : A$,
  - *pour tout* N *si* $M \rhd N$ *alors* $S^{P,A}_\Gamma(N)$, *et*
  - *pour* M_i *sous terme de* M, *il existe* B *et* Q *tel que* $\Gamma, \Delta \vdash^S M_i \xrightarrow{nf} B : Q$.

$C^C_\Gamma(A)$ *et* $C^{P,A}_\Gamma(M)$ *sont les mêmes prédicats que* $S^C_\Gamma(A)$ *et* $S^{P,A}_\Gamma(M)$ *à la différence qu'au lieu* $A \rhd B$ *et* $M \rhd N$ *nous avons* $A \rhd^+ B$ *et* $M \rhd^+ N$.

Il prouve que tous les termes et les sortes de UTT^S vérifient les deux prédicats définis précédemment, par l'intermédiaire du lemme suivant :

Lemme A.17 (Réduction)**.**

- *Si* $\Gamma \vdash^S A \xrightarrow{nf} B$ *alors* $S^B_\Gamma(A)$.
- *Si* $\Gamma \vdash^S M \xrightarrow{nf} P : A$ *alors* $S^{P,A}_\Gamma(M)$.
- *Si* $\Gamma \vdash^S M \xrightarrow{wh} N : A$ *et* $S^{P,A}_\Gamma(N)$ *alors* $S^{P,A}_\Gamma(M)$.

A partir de ce lemme, Goguen peut facilement conclure que le calcul UTT^S vérifie les propriétés désirées qui sont la normalisation forte, la préservation du type et la confluence. Il sera par la suite en mesure de transférer ces propriétés au calcul UTT en montrant que UTT est correcte par rapport à UTT^S.

A.1.1 Principe d'induction alternatif

Goguen définit un principe d'induction alternatif sur les termes et les sortes, afin de s'assurer d'avoir les prémisses correctes pour les constantes. L'idée est de connaître l'hypothèse d'induction pour les arguments qui sont des sortes pour les

constantes dont la sorte est fonctionnelle. De même que pour les constantes de type **Type**, on a besoin d'informations concernant les sortes des constructeurs de ces types. Par exemple, pour la sorte $El(Prf(\forall(A, P)))$, on a besoin de savoir que l'hypothèse d'induction est vérifiée pour la sorte $(x : El(A))El(Prf(P(x)))$, qui est un argument du constructeur Λ.
Si S est un prédicat sur les termes et Q un prédicat sur les sortes, nous avons les clauses suivantes pour le principe d'induction.

- Si Q est vrai pour **Type** et S est vrai pour M alors Q est vrai pour $El(M)$.

- Si Q est vrai pour $(Prop)$**Type** alors S est vrai pour Prf.

- Si Q est vrai pour $(A : \textbf{Type})((A)Prop)Prop$ et pour $(x : A)Prf(P(x))$ alors S est vrai pour $\forall$.

- Si Q est vrai pour le kind de $E^X[\bar{\Theta}]$, S est vrai pour $M^X[\bar{\Theta}]$, S est vrai pour $\iota_i^X[\bar{\Theta}]$ et S est vrai pour $\Phi_{i_j}^{\natural}[A, C, f, z]$, où A, C, f et z sont des variables et i_j prend la valeur des i pour les opérateurs strictement positifs, alors S est vrai pour $E^X[\bar{\Theta}]$.

Si ces clauses et celles définies pour les autres constantes sont vérifiées, alors S est vrai pour tous les termes et Q est vrai pour toutes les sortes. La validité de ce principe d'induction peut être prouvée en utilisant le principe d'induction usuel et les preuves pour les constructeurs basiques. Goguen remarque qu'un moyen simple de vérifier la validité de ce principe est de constater que la sorte de chaque constante dépend seulement des constantes définies avant elle.

A.2 Construction d'un modèle ensembliste classique

Dans ce chapitre, Goguen considère une sémantique de UTT dans la théorie classique des ensembles. Son approche est liée à la sémantique proposée par P. Dybjer [Dyb91] pour la théorie des types de Martin-Löf. Essentiellement, les types sont interprétés par des ensembles de leurs éléments, à l'exception des propositions qui sont uniquement vraies ou fausses. Les fonctions seront des fonctions ensemblistes, les entiers naturels resteront des entiers naturels, et en général les types inductifs seront les plus petits points fixes de l'équation inductive de ce type.

L'importance de ce modèle ensembliste pour la preuve de Goguen, est qu'il va justifier un autre principe d'induction sur les types qui ne pouvait être justifié syntaxiquement.

Un problème dans l'étude des systèmes aux types dépendants est que les termes, les types et les sortes sont introduits par des définitions inductives simultanément. On doit savoir ce qu'est l'environnement, avant de définir l'interpretation des termes et des sortes. Dès que l'interprétation des sortes est définie, on peut porter son attention sur les jugements dérivables et sur les environnements satisfaisant le contexte de ces jugements.

On peut interpréter la théorie des ensembles où on a admis l'existence d'un ensemble plus grand que tout ensemble, qu'on peut obtenir en utilisant des opérations ensemblistes ordinaires comme l'union et l'ensemble de tous les sous-ensembles, dans la théorie des ensembles étendue par l'axiome qui déclare l'existence d'un cardinal inaccessible. L'interprétation de Goguen suit cette intuition, en utilisant les cardinaux inaccessibles pour interpréter les univers et la sorte **Type**.

La preuve de Goguen est différente de la preuve de Dybjer pour la théorie des types de Martin-Löf, dans le sens où UTT est exprimé dans LF et que l'univers imprédicatif des propositions y est inclus. Il ne considère pas des familles inductives de types. Une modification moins importante est qu'à la place des "ensembles de règles" de Aczel [Acz86], il utilise l'induction transfinie et la construction de plus petit point fixe.

Goguen travaille essentiellement avec la théorie UTT^{-}, et pas UTT. Il justifie ce choix dans son chapitre 4 en considérant l'admissibilité des règles de substitution. Notons que nos algorithmes d'élimination des règles de substitution de la section 4.2.3 donnent une justification beaucoup plus constructive.

Dans sa construction, Goguen assume l'existence de trois cardinaux fortement inaccessibles, κ_0, κ_1 et κ_2 avec $\kappa_0 < \kappa_1 < \kappa_2$. L'environnement pour κ_2 est appelé environnement d'ensemble. L'environnement pour un ensemble S est simplement une fonction d'un ensemble fini de variables vers S.

La preuve utilise comme technique la définition d'une interprétation partielle des termes et démontre que cette interprétation est totale pour les termes bien typés.

Il faut remarquer que dans sa preuve, une autre assomption importante est l'existence des ordinaux comme représentations canoniques pour chaque cardinal κ, notamment, pour les cardinaux inaccessibles κ_0, κ_1, κ_2 mentionnés avant. Cela nécessite l'utilisation de l'axiome de choix.

On peut dire que la preuve de correction de Goguen utilise des principes assez forts, comme la cohérence (consistance) de la théorie des ensembles avec l'axiome du choix et les axiomes d'existence des cardinaux inaccessibles.

A.2.1 Grandes étapes de la preuve

Nous rappelons qu'une hiérarchie cumulative pour un ordinal α, est définie par induction sur α :

$$V_0 =_{df} \emptyset$$

$$V_{succ(\alpha} =_{df} \wp(V_\alpha)$$

$$V_\alpha =_{df} \bigcup_{\beta<\alpha} V_\beta \quad \alpha \textit{ étant une limite}$$

Cette définition peut-être trouvée dans les ouvrages standards sur la théorie des ensembles. Sa preuve définit une interprétation partielle des termes puis prouve que cette interprétation est totale pour les termes bien typés.

Dans la suite de ce chapitre, si aucune précision n'est mentionnée, κ représente un cardinal fortement inaccessible, et dans les expressions comme V_κ on assume que κ est représenté par l'ordinal correspondant. De même que α est un ordinal et ρ un environnement d'ensemble. Notons que l'expression $a \cong b$ signifie que a est défini si et seulement si b est défini et que $a = b$. Tandis que l'expression $a \succ b$ signifie que, partout où a est défini b est défini et $a = b$.

Pour un environnement d'ensemble ρ, l'interprétation d'un terme M et d'une sorte A, notée $[|M|](\rho)$ et $[|A|](\rho)$, est définie par induction sur la structure de M et A en utilisant le principe d'induction décrit à la section A.1.1. Afin de fournir une interprétation des types inductifs, il commence par définir plusieurs fonctions permettant de définir la sémantique ensembliste des types inductifs par constructions de plus petit point fixe.

- Il définit d'abord une fonction partielle sur les ensembles $\bar{\Theta}^{\#}_\rho \in V_\kappa \to V_\kappa$ par induction sur $PSCH_X(\bar{\Theta})$, qui représente une séquence de pré-schéma inductifs, pour un ordinal κ (représentant un cardinal fortement inaccessible) et un environnement d'ensemble ρ. Cet opérateur construit un ensemble inductivement.
  Certaines propriétés sont vérifiées pour la fonction $\bar{\Theta}^{\#}_\rho$ comme :

– sa monotonicité ;

– sa délimitation c'est-à-dire, en prenant un environnement d'ensemble ρ, en supposant $PSCH_X(\bar{\Theta})$ et que $\bar{\Theta}^{\#}_{\rho}$ soit défini relativement à κ, il existe un ordinal $\alpha_0 < \kappa$ tel que $\bar{\Theta}^{\#}_{\rho}(S) \in \ V_\kappa \rightarrow V_\kappa$;

– l'intervention de $\bar{\Theta}^{\#}_{\rho}$ dans l'interprétation des pré-schéma inductifs Θ pour un ensemble S, $[|\Theta|](\rho[X := S]) \cong \bar{\Theta}^{\#}_{\rho}(S) \rightarrow S$.

- Goguen définit ensuite par induction sur α, une fonction ensembliste $F \in V_\kappa \rightarrow V_\kappa$ de la manière suivante :

$$\begin{array}{c} F_0 =_{df} \emptyset \\ F_{succ(\alpha} =_{df} F(F_\alpha) \\ F_\alpha =_{df} \bigcup_{\beta<\alpha} F_\beta \quad \alpha \textit{ étant une limite.} \end{array}$$

  Puis énonce la proposition suivante :
  si $F \in \ V_\alpha \rightarrow V_\alpha$ et F est monotone, alors il existe un ordinal α_0 tel que F^{α_0} soit le plus petit point fixe pour F.

A l'aide des opérateurs introduits précédemment, il est en mesure de définir l'interprétation des types inductifs et de leurs constructeurs.

Si $\bar{\Theta}^{\#}_{\rho}(S)$ est définie relativement à un κ donné, et si α_0 est un ordinal tel que $(\bar{\Theta}^{\#}_{\rho})^{\alpha_0}$ soit le plus petit point fixe pour $\bar{\Theta}^{\#}_{\rho}$, alors les types inductifs sont interprétés de la manière suivante :

$$[|M^X[\bar{\Theta}]|](\rho) =_{df} (\bar{\Theta}^{\#}_{\rho})^{\alpha_0}(\emptyset) \quad \text{si } \kappa \leq \kappa_1$$

$$[|\mu^X[\bar{\Theta}]|](\rho) =_{df} (\bar{\Theta}^{\#}_{\rho})^{\alpha_0}(\emptyset) \quad \text{si } \kappa \leq \kappa_1$$

Goguen définit ensuite l'interprétation d'un contexte Γ, notée $[|\Gamma|]$, comme un ensemble d'environnements d'ensemble. Puis il établit les propriétés basiques de l'interprétation. Toutes ces propriétés sont prouvées à l'aide du principe d'induction de la section A.1.1. Ces propriétés sont les suivantes.

- Soit γ un environnement pour variables et un environnement d'ensemble ρ, alors $[|M|](\rho \circ \gamma) \cong [|\bar{\gamma}(M)|](\rho)$, où $\bar{\gamma}(M)$ existe si γ est une substitution.

- Soit ρ_1 et ρ_2 deux environnements d'ensemble, si pour tout $x \in FV(M)$ on a $\rho_1(x) = \rho_2(x)$ alors $[|M|](\rho_1) \cong [|M|](\rho_2)$.

- Soit $\rho = \rho_0\rho_1$ un environnement d'ensemble, alors
$$[|M|](\rho[z := [|N|](\rho_0)])) \succ [|[N/z]M|](\rho).$$

Finalement, il prouve que le système UTT^- est correcte par rapport à l'interprétation, autrement dit, que l'interprétation est totale, c'est-à-dire définie pour tous les termes et les sortes des jugements dérivables dans UTT^-. Le lemme est le suivant.

Lemme A.18 (correction de UTT^- par rapport au modèle ensembliste)**.** *Pour tout jugement dérivable dans $\Gamma \vdash^- J$, $[|\Gamma|]$ est définie, de plus :*

- *si le jugement $\Gamma \vdash^- A$**kind** est dérivable et $\rho \in [|\Gamma|]$, alors $[|A|](\rho)$ est définie ;*

- *si le jugement $\Gamma \vdash^- M : A$ est dérivable et $\rho \in [|\Gamma|]$, alors $[|A|](\rho)$ et $[|M|](\rho)$ sont définies et $[|M|](\rho) \in [|A|](\rho)$;*

- *si le jugement $\Gamma \vdash^- A = B$ est dérivable et $\rho \in [|\Gamma|]$, alors $[|A|](\rho)$ et $[|B|](\rho)$ sont définies et $[|A|](\rho) = [|B|](\rho)$;*

- *si le jugement $\Gamma \vdash^- M = N : A$ est dérivable et $\rho \in [|\Gamma|]$, alors $[|M|](\rho)$, $[|N|](\rho)$ et $[|A|](\rho)$ sont définies, $[|M|](\rho) = [|N|](\rho)$, $[|M|](\rho) \in [|A|](\rho)$ et $[|N|](\rho) \in [|A|](\rho)$.*

A.3 Preuve de la correction UTT par rapport à UTT^S

Remarque A.19. *Nous rappelons qu'une théorie T est "correcte" par rapport à une sémantique S si chaque formule dérivable dans T est vraie dans la sémantique. Ici UTT^S est la sémantique de UTT. Par exemple, le jugement $\Gamma \vdash M : A$*

dérivable dans UTT est vraie dans UTT^S si le jugement $\Gamma \vdash^S M \xrightarrow{nf} P : A$ est dérivable dans UTT^S.

De même, on parle de complétude d'une théorie T par rapport à une sémantique S, si toute formule vraie de S est dérivable dans T. Ici parler de la complétude UTT par rapport à UTT^S signifie que si par exemple le jugement $\Gamma \vdash^S M \xrightarrow{nf} P : A$ est dérivable dans UTT^S alors les jugements $\Gamma \vdash M : A$ et $\Gamma \vdash M = P : A$ sont dérivables dans UTT.

La démonstration proposée par Goguen pour montrer que UTT est correcte par rapport à UTT^S est étroitement liée aux preuves de normalisation existantes. Cependant, la preuve utilisant UTT^S, permet de rendre plus explicites les conditions exactes nécessaires pour la preuve : les règles de UTT^S montrent que les termes bien typés sont exactement ceux qui sont construits en utilisant les constructeurs canoniques et des séries d'expansions de tête faible.

Les preuves de normalisation pour les systèmes aux types dependants, proposées par exemple par Pottinger [Pot87] et Luo [Luo90a], ont utilisé un contexte infini avec un nombre infini de variables de chaque type et ont montré que chaque terme typable dans le système originel est typable dans ce contexte infini. Goguen utilise la technique de construction de modèles fondés sur les modèles de Kripke.

La technique courante pour montrer la normalisation des systèmes plus faibles, interprète les types comme des ensembles de termes pas nécessairement bien typés.

Dans sa preuve, Goguen cherche à montrer que UTT est correcte par rapport à UTT^S, qui est vue comme une sémantique, alors il est obligé de construire un modèle avec les termes bien typés.

Parmi les preuves qui n'utilisent pas des contextes infinis et interprètent les types comme des ensembles de termes bien typés, Goguen s'appuie sur Martin-Löf [ML75] et surtout Coquand [CG90] qui a proposé d'utiliser la construction des modèles de Kripke pour démontrer la normalisation forte pour un sous-ensemble des systèmes de Martin-Löf.

Sa preuve suit les techniques standards utilisées pour prouver la normalisation [Luo90a, ML72, Coq91], cependant il faut tout de même noter quelques différences qui sont les suivantes :

- Il effectue sa preuve pour le système UTT, formulé dans LF, qui contient un univers prédicatif, une large classe de types inductifs et un univers imprédicatif. L'imprédicativité l'oblige à utiliser des techniques de preuves compliquées.

- Sa preuve requiert que la notion de normalisation forte soit intégrée dans la définition de l'interprétation des sortes, comme pour Martin-Löf et Koletsos.

- Il fournit une construction ensembliste de la mesure de complexité qu'il utilise dans sa preuve.

- il prouve que UTT est correcte par rapport à UTT^S au lieu de prouver la normalisation. Cela signifie en accord avec le théorème A.11(complétude), que tous les résultats concernant UTT^S pourront être transférés à UTT, en particulier les propriétés de renforcement, de préservation du type et de confluence, en plus de la normalisation forte.

A.3.1 Les grandes étapes de la preuve

Goguen cherche tout d'abord à construire un modèle. La première préoccupation dans la construction d'un modèle est le domaine sémantique de discussion. Il commence par définir les objets sémantiques pour la preuve de correction. Comme il prouve la correction de UTT par rapport à UTT^S, les premiers composants des objets sémantiques sont les termes dérivables dans UTT^{rS}. Il montre que, relativement à n'importe quelle substitution, tous les termes typables dans UTT font partie de l'interprétation de leurs sortes, où les sortes sont interprétées comme étant les ensembles de termes bien typés dans UTT^S.

Il est bien connu que combiner le puissant principe logique de l'imprédicativité avec d'autres principes, peut facilement mener à une inconsistance. Par conséquent, un pas important pour montrer la consistance d'une théorie de types est de fournir une mesure de complexité sur les types de la théorie.

Comme nous l'avons vu, Goguen a proposé un modèle ensembliste afin d'en tirer une mesure de complexité. Intuitivement, il a besoin de fournir une interprétation ensembliste pour les types qui ne sont pas des propositions. L'idée est qu'un composant de l'interprétation est essentiellement un ensemble, mais où les types sont des ensembles de termes. Ceci permet d'utiliser les substitutions pour modéliser la quantification sur les types. Par exemple, pour interpréter la proposition imprédicative $\forall(Prop, [X : Prop]X)$ qui quantifie sur les propositions, il faut être en mesure de considérer des interprétations arbitraires possibles des propositions.

L'interprétation du quantificateur sera l'intersection de toutes ces interprétations possibles. Il en aura besoin non seulement pour les propositions mais pour tous les types de la théorie à l'exclusion des sortes. Ce problème a motivé la définition des ensembles valués qui caractérisent le comportement ensembliste des éléments de chaque type.

Goguen n'utilise pas la notion de quasinormalisation proposée par Luo, pour donner une mesure de complexité pour la théorie des types ECC. En revanche, il définit une mesure de complexité en considérant la complexité de l'interprétation ensembliste, ce qui ne dépend pas de la forme normale des types. Il utilise cette mesure de complexité, non à cause de la difficulté à obtenir des formes normales mais plutôt parce qu'il semble qu'il n'y ait pas de mesure simple pour les types de UTT.

Afin d'élaborer une interprétation de la théorie des types à la manière des modèles de Kripke, Goguen a besoin que les valeurs sémantiques soient fermées par rapport aux transitions, lesquelles dans son cas, sont les extensions de contextes δ de Δ' vers Δ. Or on sait que l'ensemble des termes typés dans UTT^S est fermé par rapport aux extensions de contextes, lemme A.14 (Renommage)à l'appui. Pour permettre aux objets sémantiques qui contiennent des valeurs, d'être fermés de manière similaire, il a indexé les valeurs par des extensions de contextes. Les objets sémantiques pour un contexte Δ sont des paires composées d'un terme bien typé dans UTT^S dans le contexte Δ et d'une famille de valeurs indexée par des extensions de contextes δ de Δ' vers Δ.

L'égalité que veut établir Goguen pour les objets sémantiques, est la suivante : deux objets sémantiques sont égaux si les termes qui les composent ont la même forme normale et si les valeurs indexées qui les composent sont égales pour toute extension. Il montre également que la règle d'application ($App - Eq$) est correcte, dans le sens où si (M, v_M) est un objet sémantique pour $(x : A_1)A_2$ dans le contexte Δ et que (N_1, v) et (N_2, v) sont deux objets sémantiques égaux pour A_1 dans le contexte Δ, alors on a $APP_\Delta((M, v_M), (N_1, v)) = APP_\Delta((M, v_M), (N_2, v))$. Cela implique que la fonction v_M retourne une valeur qui dépend seulement de la forme normale du terme qui compose l'objet sémantique placé en argument. Plusieurs preuves de normalisation incluent cette condition dans la définition des valeurs, mais il suit la méthode de Coquand en donnant cette condition comme un prédicat sur les objets sémantiques ; voir la définition sur les objets sémantiques uniformes.

De manière plus formelle, sans entrer ici dans tous les détails, Goguen définit les notions vues précédemment de la façon suivante :

- Un objet sémantique pour un contexte Δ et une sorte A, est un couple (M, v), où $\Delta \vdash^S M \xrightarrow{nf} P : A$ et v est une famille d'objets $v(\delta) \in V(\Delta' \vdash^S \widehat{\delta}(M) : \widehat{\delta}(M))$, indexés par des renommages δ de Δ' vers Δ. $V(\Delta \vdash^S M : A)$, où $\Delta \vdash^S M \xrightarrow{nf} M : A$ est définie par induction sur la complexité de A.
Il faut noter que :

  - l'ensemble des objets sémantiques pour Δ et A est noté $SO_\Delta(A)$;

  - S est un ensemble saturé pour Δ et A, *notation* $SAT_\Delta(A)$, si S est un ensemble d'objets sémantiques pour Δ et A tel que :

    * **(S_1)** si M est un terme de base tel que $\Delta \vdash^S M \xrightarrow{nf} P : A$ alors $(M, \lambda\delta.*)$ est dans S, et

    * **(S_2)** si (N, v) est dans S et $\Delta \vdash^S M \xrightarrow{wh} N : A$ alors (M, v) est dans S.

Nous listons quelques remarques importantes par rapport à cette définition :

- Les objets sémantiques ont des éléments toujours fortement normalisants.

- Dans beaucoup de preuve de normalisation forte pour les calculs contenant l'imprédicativité, on retrouve un lemme disant que pour tout contexte Δ, terme M et sorte A, il existe une valeur canonique $v_{\Delta,M,A} \in V(\Delta \vdash^S M : A)$. Goguen n'a pas besoin de ce lemme car dans sa formulation, il est évident que si $\Delta \vdash^S M \xrightarrow{nf} P : A$ et que P est un terme de base alors $\in\ = V(\Delta \vdash^S P : A$.

- Le domaine d'une fonction dans l'ensemble valué pour une sorte dépendante $(x : A_1)A_2$ n'a pas besoin d'être tout $SO_\Delta(A_1)$. Quand on définit par exemple l'interprétation de $[x : A_1]M_0$, le domaine de la fonction définie doit être l'interprétation de A_1, car les hypothèses d'induction s'appliqueront uniquement aux éléments de cet ensemble. C'est la raison pour laquelle les termes de base ont le même ensemble valué, peu importe leur sorte, car on ne connaît pas à l'avance l'interprétation de A_1, mais nous devons savoir qu'un terme de base est toujours dans les ensembles saturés.

- A cause du lemme d'unicité des formes normales, on n'est pas en mesure de différencier un terme M bien typé dans UTT^S dans un contexte Δ et une preuve de $\Delta \vdash^S M \xrightarrow{nf} P : A$.

- Si (M, v_M) est un objet sémantique pour Δ et A, et δ un renommage de Δ' vers Δ, alors la fonction monotonie est :
$$mon_\delta(M, v_M) =_{df} (\widehat{\delta}(M), \lambda\delta'.v_M(\delta' \circ \delta))$$
Par construction on sait que $mon_\delta(M, v_M)$ est un objet sémantique pour Δ' et A.

- Soit (M_1, v_1) un objet sémantique pour Δ et $(x : A_1)A_2$ tel que $\Delta \vdash^S M_1 \xrightarrow{nf} P_1 : (x : A_1)A_2$ et (M_2, v_2) est un objet sémantique pour Δ et A_1 tel que $\Delta \vdash^S M_2 \xrightarrow{nf} P_2 : A_1$. Nous montrons qu'il existe une sorte B_2 telle que $\Delta \vdash^S [M_2/x]A_2 \xrightarrow{nf} B_2$ et définissons l'application
$$APP_\Delta((M_1, v_1), (M_2, v_2)) \in SO_\Delta(B_2)$$
Il faut considérer la notation $(M, v_M) \downarrow (N, v_N)$ où (M, v_M) et (N, v_N) sont des objets sémantiques pour Δ et A, s'il existe un terme P tel que $\Delta \vdash^S M \downarrow N[P] : A$ et $v_M = v_N$.

- Soit S une famille saturée d'ensembles pour Δ et A. S est uniforme si $(M, v_M) \in S(\delta)$ implique que (M, v_M) est uniforme et $mon_{\delta'}(M, v_M) \in S(\delta' \circ \delta)$ pour tout renommage δ' de Δ'' vers Δ'.

  Voici quelques propriétés simples des objets sémantiques uniformes :

  - $USO_\Delta(A)$ est uniforme et est dans $SAT_\Delta(A)$;

  - si (M, v_M) est un objet sémantique uniforme alors $mon_\delta(M, v_M)$ est aussi un objet sémantique uniforme, pour tout renommage δ de Δ' vers Δ.

Goguen poursuit en introduisant la notion d'environnement pour les objets sémantiques. Il fournit les définitions suivantes :

- Une pré-substitution est un environnement pour termes.

- Une substitution ϕ de Δ vers Γ, où $\Gamma \equiv x_1 : A_1, \ldots, x_m : A_m$ et $\Delta \vdash^S$ **valid**, est une pré-substitution sur Γ telle que :
$\Delta \vdash^S \widehat{\phi}(A_i) \xrightarrow{nf} B_i$ et $\Delta \vdash^S \phi(x_i) \xrightarrow{nf} P : B_i$ pour $1 \leq i \leq m$.

- Une pré-valuation est un environnement pour objets sémantiques dans un contexte donné Δ.

- Une valuation de Δ vers Γ est une pre-valuation telle que le terme constituant est une substitution de Δ vers Γ.

- Une valuation ρ de Δ vers Γ est uniforme si $\rho(x)$ est uniforme pour tout $(x : A) \in \Gamma$.

Une fois les objets sémantiques définis, Goguen suit la technique de Streicher en donnant une interprétation partielle des termes par des objets sémantiques et des sortes par des ensembles d'objets sémantiques relative à un environnement d'objets sémantiques.
L'interprétation d'un terme M, notée $[|M|]\rho\Delta$ et l'interprétation d'une sorte A, notée $[|A|]\rho\Delta$, relative à une valuation ρ de Δ vers Γ sont définies par induction sur la structure de M et de A.
$[|M|]\rho\Delta$ est un couple $(\widehat{\phi}(M), V[|M|]\rho\Delta)$ où $V[|M|]\rho\Delta$ est une famille de valeurs indexée par un renommage δ' de Δ' vers Δ.

Cependant, pour montrer que l'interprétation est un objet sémantique, si elle est définie, il montre simultanément la distributivité de la fonction monotonicité et de l'interprétation, à savoir :

$$APP''_{\Delta}(mon_{\delta' \circ \delta}(M, v_M), mon_{\delta'}(N, v_N)) =$$
$$mon_{\delta'}(APP_{\Delta'}(mon_{\delta}(M, v_M), (N, v_N)))$$

avec δ un renommage de Δ' vers Δ et δ' un renommage de Δ'' vers Δ'.

Il définit également l'interprétation pour les contextes comme des environnements, dont les objets sémantiques sont des éléments de l'interprétation des sortes qui constituent le contexte.

Goguen vérifie ensuite que l'interprétation possède les propriétés désirées. Il prouve par induction sur la structure des termes des lemmes concernant :

- le changement des variables liées,

- l'égalité de l'interprétation sous des environnements qui correspondent au variables libres,

- l'uniformité de l'interprétation, pour la correction de la règle d'application de l'égalité $(App - Eq)$, et permettre de passer d'une valuation dans un contexte Δ à une valuation dans un contexte Δ' pour un renommage de Δ à Δ' donné, et

- une propriété des substitutions, pour l'interprétation utilisée dans la correction des règles $(S - App)$ et (β).

Finalement, il fait la preuve finale de correction par induction sur les dérivations de UTT. Ce résultat est valable pour toute valuation dans l'interprétation du contexte, donc il prouve que si $\Gamma \vdash$ **valid** alors il existe une valuation de Γ vers Γ, qui fournit exactement le résultat de correction dont on a besoin.
Le lemme de correction est le suivant :

Lemme A.20 (Correction)**.**

- *Si* $\Gamma \vdash A$***kind*** *est dérivable alors il existe* B *tel que* $\Gamma \vdash^S A \xrightarrow{nf} B$ *soit dérivable.*

- *Si* $\Gamma \vdash M : A$ *est dérivable alors il existe* B *et* P *tels que* $\Gamma \vdash^S A \xrightarrow{nf} B$ *et* $\Gamma \vdash^S M \xrightarrow{nf} P : B$ *soient dérivables.*

- *Si* $\Gamma \vdash A = B$ *est dérivable alors il existe* C *tel que* $\Gamma \vdash^S A \downarrow B[C]$ *soit dérivable.*

- *Si* $\Gamma \vdash M = N : A$ *est dérivable alors il existe* B *et* P *tels que* $\Gamma \vdash^S A \xrightarrow{nf} B$ *et* $\Gamma \vdash^S M \downarrow N[P] : B$ *soient dérivables.*

Ce lemme nous permet de transférer les propriétés de UTT^S à UTT, donc de déduire que les propriétés de normalisation forte, de préservation de type et de confluence sont vérifiées pour UTT.

A.4 Mesure de complexité pour UTT^-

Dans l'annexe de sa thèse, Goguen justifie le principe d'induction sur les types de UTT^-, en proposant une mesure de complexité. Ce principe joue un rôle important dans la preuve de normalisation car il permet de définir le rang de l'interprétation que Goguen donne dans son chapitre 6. Ce rang est différent de celui de l'interprétation ensembliste où, par exemple, le rang de l'interprétation ensembliste de $\Gamma \vdash^- M : A$ est juste l'univers des ensembles. Goguen utilise une variante du modèle ensembliste pour fournir une mesure de complexité. Intuitivement dans ce modèle, le rang de l'interprétation du type composant un type inductif est nécessairement inférieur à celui de ce type inductif. Cependant ceci est vrai uniquement en l'absence de types vides. Goguen considère donc une interprétation, où les propositions sont toujours non vides et restreint la syntaxe des types inductifs pour proscrire les types vides.

Goguen définit donc de manière standard le rang d'un ensemble. Il faut noter cependant la nécessité de l'axiome de fondation pour cette définition.

Définition A.21. *Le rang d'un ensemble S est le plus petit ordinal plus grand que le rang de tous les éléments de S.*

$$rank(S) = min\{\omega | rank(T) < \omega, T \in S\}$$

$$\textit{avec } rank(\emptyset) = 0$$

Les propriétés suivantes sont vérifiées pour la fonction $rank(S)$.

- Si A et B sont des ensembles alors $max rank(A), rank(B) \leq rank(A + B)$.
- Si A et B sont des ensembles non vides alors $max rank(A), rank(B) < rank(A \times B)$.
- Si A est un ensemble non vide et B une famille d'ensembles non vides sur A alors $max(\{rank(A)\} \cup \{rank(B(a)) | a \in A\}) < rank(Fa \in A.B(a))$, pour $F \in \{\sum, \prod\}$.

La sémantique ensembliste du chapitre A.2 interprète les propositions fausses par l'ensemble vide. Afin de n'interpréter aucun type par l'ensemble vide, Goguen propose une sémantique alternative simple dans laquelle il n'existe qu'une seule proposition avec une preuve unique. C'est simplement une interprétation inconsistante des propositions.

Goguen introduit le calcul UTT^* qui est un sous-langage de UTT^- qui ne contient pas de types vides. Il introduit les définitions suivantes :

Définition A.22 (Schéma non vide)**.** *Soit* Γ *un contexte valide. On dit que* $\bar{\Theta}$ *est non* Γ*-schéma non vide,* notation $SCH^*_{\Gamma;X}(\bar{\Theta})$ *s'il existe un* Θ_i *dont l'arité est* 0*.*

Définition A.23 (UTT^*)**.** *Le calcul* UTT^* *est le calcul* UTT^- *dans lequel toutes les occurrences de* $SCH^-_{\Gamma;X}(\bar{\Theta})$ *sont remplacées par* $SCH^*_{\Gamma;X}(\bar{\Theta})$*. Le symbole* $\vdash^*$ *est utilisé pour les jugements de* UTT^**.*

Il prouve également le lemme :

Lemme A.24. *Si* $SCH^*_{\Gamma;X}(\bar{\Theta})$ *et* S *est une ensemble non vide alors pour tout* $(\Gamma, \Gamma'; A) \in TYPES(\bar{\Theta})$ *et pour tout* $(\rho, \rho') \in [|\Gamma, \Gamma'|]$*,* $rank([|A|](\rho, \rho')) < rank(\bar{\Theta}^\sharp_\rho(S))$*.*

Goguen peut donc définir sa mesure de complexité de la façon suivante :

Définition A.25. *On définit la complexité d'un jugement dérivable* $\Gamma \vdash^* A$***kind****,* $c(\Gamma \vdash^* A\textbf{kind})$ *de la manière suivante :*

$$c(\Gamma \vdash^* A\textbf{kind}) =_{df} \mu i.(\forall \rho \in [|\Gamma|].rank([|A|](\rho)) \leq i)$$

$c(\Gamma \vdash^* A\textbf{kind})$ est un ordinal canonique.

Définition A.26 (Complexité inductive)**.** *La complexité de* M *telle que* $\Gamma \vdash^- M : M^X[\bar{\Theta}]$*, est définie de la manière suivante :*

$$k(\Gamma \vdash^* M : M^X[\bar{\Theta}]) =_{df} \mu\alpha.\forall\rho.[|M|](\rho) \in (\bar{\Theta}^\sharp_\rho)^\alpha$$

$k(\Gamma \vdash^* M : M^X[\bar{\Theta}])$ est un ordinal canonique.

Annexe B

Les règles de $UTT^r[R]$

B.1 Le système $UTT^r[R]$

Contexte et assomption

$$\frac{}{<>\vdash \textbf{valid}}\ (1.1) \qquad \frac{\Gamma \vdash K\textbf{kind} \quad x \notin FV(\Gamma)}{\Gamma, x : K \vdash \textbf{valid}}\ (1.2) \qquad \frac{\Gamma, x : K, \Gamma' \vdash \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K}\ (1.3)$$

$$\frac{\Gamma_1, \Gamma_2 \vdash J \qquad \Gamma_1, \Gamma_3 \vdash \textbf{valid}}{\Gamma_1, \Gamma_3, \Gamma_2 \vdash J}\ (\text{wkn}) \quad (FV(\Gamma_2) \cap FV(\Gamma_3) = \emptyset)$$

Égalité générale

$$\frac{\Gamma \vdash K\textbf{kind}}{\Gamma \vdash K = K}\ (2.1) \qquad \frac{\Gamma \vdash K = K'}{\Gamma \vdash K' = K}\ (2.2) \qquad \frac{\Gamma \vdash K = K' \qquad \Gamma \vdash K' = K''}{\Gamma \vdash K = K''}\ (2.3)$$

$$\frac{\Gamma \vdash k : K}{\Gamma \vdash k = k : K}\ (2.4) \qquad \frac{\Gamma \vdash k = k' : K}{\Gamma \vdash k' = k : K}\ (2.5) \qquad \frac{\Gamma \vdash k = k' : K \qquad \Gamma \vdash k' = k'' : K}{\Gamma \vdash k = k'' : K}\ (2.6)$$

Retypage

$$\frac{\Gamma \vdash k : K \qquad \Gamma \vdash K = K'}{\Gamma \vdash k : K'}\ (3.1) \qquad \frac{\Gamma \vdash k = k' : K \qquad \Gamma \vdash K = K'}{\Gamma \vdash k = k' : K'}\ (3.2)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash J \qquad \Gamma \vdash K = K'}{\Gamma, x : K', \Gamma' \vdash J}\ (3.3)$$

Le sorte Type

$$\frac{\Gamma \vdash \textbf{valid}}{\Gamma \vdash \textbf{Type kind}}\ (4.1) \qquad \frac{\Gamma \vdash A : \textbf{Type}}{\Gamma \vdash El(A)\textbf{kind}}\ (4.2) \qquad \frac{\Gamma \vdash A = B : \textbf{Type}}{\Gamma \vdash El(A) = El(B)\textbf{kind}}\ (4.3)$$

Produit dépendant

$$\frac{\Gamma, x : K \vdash K'\textbf{kind}}{\Gamma \vdash (x : K)K'\textbf{kind}}\ (5.1) \qquad \frac{\Gamma, x : K_1 \vdash K_1' = K_2' \qquad \Gamma \vdash K_1 = K_2}{\Gamma \vdash (x : K_1)K_1' = (x : K_2)K_2'}\ (5.2)$$

$$\frac{\Gamma, x : K \vdash k : K'}{\Gamma \vdash [x : K]k : (x : K)K'}\ (5.3) \qquad \frac{\Gamma, x : K_1 \vdash k_1 = k_2 : K \qquad \Gamma \vdash K_1 = K_2}{\Gamma \vdash [x : K_1]k_1 = [x : K_2]k_2 : (x : K_1)K}\ (5.4)$$

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k : K}{\Gamma \vdash f(k) : [k/x]K'}\ (5.5) \qquad \frac{\Gamma \vdash f = f' : (x : K)K' \qquad \Gamma \vdash k_1 = k_2 : K}{\Gamma \vdash f(k_1) = f'(k_2) : [k_1/x]K'}\ (5.6)$$

$$\frac{\Gamma, x : K \vdash k' : K' \qquad \Gamma \vdash k : K}{\Gamma \vdash ([x : K]k')(k) = [k/x]k' : [k/x]K'}\ (5.7) \qquad \frac{\Gamma \vdash f : (x : K)K' \quad x \notin FV(\Gamma)}{\Gamma \vdash [x : K]f(x) = f : (x : K)K'}\ (5.8)$$

Substitutions

$$\frac{\Gamma, x : K, \Gamma' \vdash \textbf{valid} \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash \textbf{valid}}\ (6.1) \qquad \frac{\Gamma, x : K, \Gamma' \vdash K'\textbf{kind} \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]K'\textbf{kind}}\ (6.2)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash k' : K' \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]k' : [k/x]K'\textbf{kind}}\ (6.3) \qquad \frac{\Gamma, x : K, \Gamma' \vdash K' = K'' \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]K' = [k/x]K''}\ (6.4)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash k' = k'' : K' \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]k' = [k/x]k'' : [k/x]K'}\ (6.5)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash K'\textbf{kind} \qquad \Gamma \vdash k_1 = k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]K' = [k_2/x]K'}\ (6.6)$$

$$\frac{\Gamma, x : K, \Gamma' \vdash k' : K' \qquad \Gamma \vdash k_1 = k_2 : K}{\Gamma, [k_1/x]\Gamma' \vdash [k_1/x]k' = [k_2/x]k' : [k_1/x]K'}\ (6.7)$$

Types inductifs

$$\frac{PSCH_X(\bar{\Theta}) \qquad \Gamma, X : \textbf{Type} \vdash \Theta_i\textbf{kind}}{\Gamma \vdash M^X[\bar{\Theta}] : \textbf{Type}}\ \text{(C-M)} \quad 1 \leq i \leq n \text{ et } \bar{\Theta} \text{ de longueur } n$$

$$\frac{PSCH_X(\bar{\Theta}) \qquad \Gamma, X : \textbf{Type} \vdash \Theta_i \textbf{kind}}{\Gamma \vdash \iota_i^X[\bar{\Theta}] : \Theta_i(M^X[\bar{\Theta}])} \text{ (C-i)} \quad 1 \leq i \leq n \text{ et } \bar{\Theta} \text{ de longueur } n$$

$$(\kappa^X\text{-EQ}) \; \frac{\begin{array}{c} SCH_{\Gamma,X}(\bar{\Theta}) \quad SCH_{\Gamma,X}(\bar{\Theta}') \qquad \Gamma \vdash \kappa^X[\bar{\Theta}] : K \quad \Gamma \vdash \kappa^X[\bar{\Theta}'] : K \\ \Gamma, X : \textbf{Type} \vdash \bar{\Theta}_i = \bar{\Theta}_i' \quad (i = 1, \ldots, n) \end{array}}{\Gamma \vdash \kappa^X[\bar{\Theta}] = \kappa^X[\bar{\Theta}'] : K}$$

$$\frac{PSCH_X(\bar{\Theta}) \quad \Gamma, X : \textbf{Type} \vdash \Theta_i \textbf{kind} \quad 1 \leq i \leq n}{\Gamma \vdash E^X[\bar{\Theta}] : (C : (M^X[\bar{\Theta}])\textbf{Type})} \text{ (C-E)}$$

$$(f_1 : \Theta_1^\circ[M^X[\bar{\Theta}], C, \iota_1^X[\bar{\Theta}]])$$

$$\ldots$$

$$(f_n : \Theta_n^\circ[M^X[\bar{\Theta}], C, \iota_n^X[\bar{\Theta}]])(z : M^X[\bar{\Theta}])C(z)$$

$$\frac{\begin{array}{c} PSCH_X(\bar{\Theta}) \qquad \Gamma, X : \textbf{Type} \vdash \Theta_i \textbf{kind} \\ \Gamma \vdash C : (M^X[\bar{\Theta}])\textbf{Type} \qquad \Gamma \vdash f_i : \Theta_i^\circ[M^X[\bar{\Theta}], C, \iota_i^X[\bar{\Theta}]] \\ \Gamma \vdash a_j : M_j(M^X[\bar{\Theta}]) \quad 1 \leq j \leq m \quad 1 \leq i \leq n \end{array}}{\Gamma \vdash E^X[\bar{\Theta}](C, \bar{f}, \iota_i^X[\bar{\Theta}](\bar{a})) = f_i(\bar{a}, \Phi_1^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_1], \ldots, \Phi_k^\natural[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_k]) : C(\iota_i^X[\bar{\Theta}](\bar{a}))} \text{ (E-EQ)}$$

Avec $\bar{f}$ qui représente $f_1, \ldots, f_n$, $\bar{a}$ qui représente $a_1, \ldots, a_m$ et $ARITY_X(\Theta) \equiv \Phi_1, \ldots, \Phi_k$. Rappelons que Θ_i est de la forme $(x_1 : M_1) \ldots (x_m : M_m)X$.

Nouvelle égalité

$$\frac{\Gamma \vdash E[\bar{\Theta}_m](C, a_1, ..., a_m) : (F_m)A \qquad \Gamma \vdash E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m]) : (F_n)F_m \qquad \Gamma \vdash k : F_n}{\Gamma \vdash E[\bar{\Theta}_m](C, a_1, ..., a_m)(E[\bar{\Theta}_n](C_1, \iota_{\sigma(1)}^X[\bar{\Theta}_m], ..., \iota_{\sigma(n)}^X[\bar{\Theta}_m])(k)) = E[\bar{\Theta}_n](C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A} \text{ (ϑ-eq)}$$

Avec :
$C \equiv [y : F_m]A : (y : F_m)$**Type**,
$C_1 \equiv [z : F_n]F_m : (z : F_n)$**Type**,
$C_2 \equiv [x : F_n]A : (x : F_n)$**Type**.

Univers imprédicatif

$$Prop : \mathbf{Type}$$

$$\mathbf{Prf} : (Prop)\mathbf{Type}$$

$$\forall : (A : \mathbf{Type})((A)Prop)Prop$$

$$\Lambda : (A : \mathbf{Type})(P : (A)Prop)((x : A)\mathbf{Prf}(P(x)))\mathbf{Prf}(\forall(A, P))$$

$$\mathbf{E}_\forall : (A : \mathbf{Type})(P : (A)Prop)(R : \mathbf{Prf}(\forall(A, P)))Prop)((g : (x : A)\mathbf{Prf}(P(x)))\mathbf{Prf}(R(\Lambda(A, P, g))))(z : \mathbf{Prf}(\forall(A, P)))\mathbf{Prf}(R(z))$$

On a l'égalité calculatoire suivante :

$$\mathbf{E}_\forall(A, P, R, f, \Lambda(A, P, g)) = f(g) : \mathbf{Prf}(R(\Lambda(A, Pg))).$$

Univers prédicatif

$$\mathbf{U} : \mathbf{Type}$$

$$\mathbf{T} : (\mathbf{U})\mathbf{Type}$$

$$prop : \mathbf{U}$$

$$\mu^X[\bar{\Theta}] : \mathbf{U}$$

$$prf : (Prop)\mathbf{U}$$

Les règles d'égalité associées sont les suivantes :

$$\frac{\Gamma \vdash \textbf{valid}}{\Gamma \vdash \mathbf{T}(prop) = Prop : \textbf{Type}}\ \text{(T-prop)} \qquad \frac{\Gamma \vdash P : Prop}{\Gamma \vdash \mathbf{T}(prf(P)) = \mathbf{Prf}(P) : \textbf{Type}}\ \text{(T-prf)}$$

$$\frac{\Gamma \vdash \mu^X[\bar{\Theta}] : \mathbf{U}}{\Gamma \vdash \mathbf{T}(\mu^X[\bar{\Theta}]) = M^X[\bar{\Theta}] : \textbf{Type}}\ \text{(T-}\mu\text{)}$$

Sous-typage

- **Coercions de base**

$$\frac{\Gamma \vdash A : \textbf{Type} \quad \Gamma \vdash B : \textbf{Type} \qquad \Gamma \vdash c : (El(A))El(B) \quad (A, B, c) \in R}{\Gamma \vdash A <_c B : \textbf{Type}}\ \text{(ST.1)}$$

- **Transitivité et congruence**

$$\frac{\Gamma \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash A = A' : \textbf{Type}}{\Gamma \vdash A' <_c B : \textbf{Type}}\ \text{(ST.2)}$$

$$\frac{\Gamma \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash B = B' : \textbf{Type}}{\Gamma \vdash A <_c B' : \textbf{Type}}\ \text{(ST.3)}$$

$$\frac{\Gamma \vdash A <_{c_1} B : \textbf{Type} \qquad \Gamma \vdash B <_{c_2} C : \textbf{Type}}{\Gamma \vdash A <_{[x:El(A)]c_2(c_1(x))} C : \textbf{Type}}\ \text{(ST.4)}$$

$$\frac{\Gamma \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash c = c' : (El(A))El(B)}{\Gamma \vdash A <_{c'} B : \textbf{Type}}\ \text{(ST.5)}$$

- **Substitutions**

$$\frac{\Gamma, x : K, \Gamma' \vdash A <_c B : \textbf{Type} \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]A <_{[k/x]c} [k/x]B : \textbf{Type}}\ \text{(ST.6)}$$

Sous-sortes

- **Règles basiques**

$$\frac{\Gamma \vdash A <_c B : \textbf{Type}}{\Gamma \vdash El(A) <_c El(B) : \textbf{Type}} \text{ (SK.1)}$$

- **Sous-sortes et produit dépendant**

$$\frac{\Gamma \vdash K_1' <_{c_1} K_1 \qquad \Gamma, x' : K_1' \vdash [c_1(x')/x]K_2 = K_2' \qquad \Gamma, x : K_1 \vdash K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K_1')K_2'} \text{ (SK.2)}$$

avec $c \equiv [f : (x : K_1)K_2][x' : K_1']f(c_1(x'))$.

$$\frac{\Gamma \vdash K_1' = K_1 \qquad \Gamma, x' : K_1' \vdash K_2 <_{c_2} K_2' \qquad \Gamma, x : K_1 \vdash K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K_1')K_2'} \text{ (SK.3)}$$

avec $c \equiv [f : (x : K_1)K_2][x' : K_1']c_2(f(x))$.

$$\frac{\Gamma \vdash K_1' <_{c_1} K_1 \qquad \Gamma, x' : K_1' \vdash [c_1(x')/x]K_2 <_{c_2} K_2' \qquad \Gamma, x : K_1 \vdash K_2\textbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_c (x' : K_1')K_2'} \text{ (SK.4)}$$

avec $c \equiv [f : (x : K_1)K_2][x' : K_1']c_2(f(c_1(x')))$.

- **Transitivité et congruence pour sous-sortes**

$$\frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_2 = K_2'}{\Gamma \vdash K_1 <_c K_2'} \text{ (SK.5)} \qquad \frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_1 = K_1'}{\Gamma \vdash K_1' <_c K_2} \text{ (SK.6)}$$

$$\frac{\Gamma \vdash K <_{c_1} K' \qquad \Gamma \vdash K' <_{c_2} K''}{\Gamma \vdash K <_{[x:K]c_2(c_1(x))} K''} \text{ (SK.7)} \qquad \frac{\Gamma \vdash K <_c K' \qquad \Gamma \vdash c = c' : (K)K'}{\Gamma \vdash K <_{c'} K'} \text{ (SK.8)}$$

- **Substitution pour sous-sortes**

$$\frac{\Gamma, x : K, \Gamma' \vdash K_1 <_c K_2 \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]K_1 <_{[k/x]c} [k/x]K_2} \text{ (SK.9)}$$

Règles coercitives

- **Application coercitive**

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k) : [c(k)/x]K'} \text{ (CA.1)}$$

$$\frac{\Gamma \vdash f = f' : (x : K)K' \qquad \Gamma \vdash k_1 = k_2 : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_1) = f'(k_2) : [c(k_1)/x]K'} \text{ (CA.2)}$$

- **Définition coercitive**

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k_0 : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_0) = f(c(k_0)) : [c(k_0)/x]K'} \text{ (CD)}$$

Bibliographie

[AC96] D. Aspinall and A. Compagnoni. Subtyping dependent types. *Proc. of LICS96*, 1996.

[ACN90] L. Augustsson, Th. Coquand, and B. Nordström. A short description of another logical framework. In G. Huet and G. Plotkin, editors, *Preliminary Proc. of Logical Frameworks*, 1990.

[Acz86] P. Aczel. The type theoretic interpretation of constructive set theory : inductive definitions. In *Logic, Methodology and Philosophy of Science VII*, 1986.

[Acz94] P. Aczel. Simple overloading for type theories. Draft, 1994.

[Acz98] Peter Aczel. On relating type theories and set theories. In *Proceedings of TYPES'98, volume 1657 of Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 1998.

[AGM92] S. Abramsky, D. Gabbay, and T. Maibaum. *Handbook of Logic in Computer Science*, volume 2. Clarendon Press, 1992.

[AHMP92] Arnon Avron, Furio Honsell, Ian Mason, and Robert Pollack. Using typed lambda calculus to implement formal systems on a machine. *Journal of Automated Reasoning*, 9 :309–352, 1992.

[Alt] Th. Altenkirch. *Constructions, Inductive Types and Strong Normalization*. PhD thesis, Edinburgh University. forthcoming.

[Alt92] Th. Altenkirch. Brewing strong normalization proofs with LEGO. LFCS report ECS-LFCS-92-230, Dept. of Computer Science, Edinburgh University, 1992.

[Alt93a] Th. Altenkirch. A formalization of the strong normalization proof for System F in LEGO. In J.F. Groote M. Bezem, editor, *Typed Lambda Calculi and Applications*, LNCS 664, 1993.

[Alt93b] Th. Altenkirch. Yet another strong normalization proof for the Calculus of Constructions. In *Proceedings of the Vintermötet*, 1993. To appear as Chalmers Tech-report.

[And86] P.B. Andrews. *An Introduction to Mathematical Logic and Type Theory : to truth through proof.* Academic Press, 1986.

[Asp88] A. Asperti. The internal model of polymorphic lambda calculus. Report CMU-CS-88-155, Computer Science Dept., CMU, 1988.

[ASW94] H. Andersen, C. Stirling, and G. Winskel. A compositional proof system for modal μ-calculus. *LICS*, 1994.

[BA90] M. Ben-Ari. *Principles of concurrent and distributed programming*. Prentice-Hall, 1990.

[Bac88a] R. Backhouse. An exploration of the Bird-Meertens formalism. manuscript, 1988.

[Bac88b] R. Backhouse. On the meaning and construction of the rules in 's theory of types. In A. Avron , editor, *Workshop on General Logic*. LFCS Report Series, ECS-LFCS-88-52, Dept. of Computer Science, University of Edinburgh, 1988.

[Bai96] A. Bailey. Lego with implicit coercions. 1996. Draft.

[Bai98] A. Bailey. *The Machine-checked Literate Formalisation of Algebra in Type Theory*. PhD thesis, University of Manchester, 1998.

[Bar84] H.P. Barendregt. *The Lambda Calculus : its Syntax and Semantics*. North-Holland, revised edition, 1984.

[Bar89a] H.P. Barendregt. Introduction to generalized type systems. Proc. of the 3rd Italian Conf. on Theoretical Computer Science, Mandera., 1989.

[Bar89b] H.P. Barendregt. Typed lambda calculi. to appear in Handbook of Logic in Computer Science (eds., S. Abramsky, D. Gabbay and T.S.E. Maibaum) Oxford University Press, 1989.

[Bar92a] H.P. Barendregt. Lambda calculi with types. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2. Clarendon Press, 1992.

[Bar92b] H.P. Barendregt. Lambda calculi with types. In [AGM92], 1992.

[Bar96] G. Barthes. Implicit coercions in type systems. *Proceedings of Types'95, LNCS 1128*, 1996.

[BB85] C. Böhm and A. Berarducci. Automatic synthesis of typed λ-programs on term algebras. *Theoretical Computer Science*, 39, 1985.

[BCGS89] V. Breazu-Tannen, T. Coquand, C. Gunter, and A. Scedrov. Inheritance and explicit coercion. In *Proc. of LICS'89*, 1989.

[BCGS91] V. Breazu-Tannen, T. Coquand, C. Gunter, and A. Scedrov. Inheritance and explicit coercion. *Information and Computation*, 93, 1991.

[BCM89] R. Backhouse, P. Chisholm, and G. Malcolm. Do-it-youself type theory. *Formal Aspects of Computing*, 1(1), 1989.

[BD93] R. Burstall and R. Diacanescu. Hiding and behaviour : an institutional approach. to appear as invited paper in a Festschrift, Prentice Hall, 1993.

[Bee82] M.J. Beeson. Problematic principles in constructive mathematics. *Logic Colloquium'80*, 1982.

[Bee85] M.J. Beeson. *Foundations of Constructive Mathematics*. Springer-Verlag, 1985.

[Ber89a] S Berardi. Non-conservativity of coquand's calculus with respect to higher-order intuitionistic logic. Talk given in the 3rd Jumelage meeting on Typed Lambda Calculi, Edinburgh, 1989.

[Ber89b] S. Berardi. Type dependence and constructive mathematics. manuscript, June 1989.

[Ber90] S. Berardi. *Type Dependence and Constructive Mathematics*. PhD thesis, Universita di Torino, Italy, 1990.

[BG80] R. Burstall and J. Goguen. The semantics of Clear, a specification language. *Lecture Notes in Computer Science*, 86, 1980.

[BG89] E. Barendsen and H. Geuvers. Conservativity of λp over pred. manuscript, 1989.

[Bir87] R. Bird. An introduction to the theory of lists. In M. Broy, editor, *Logic of programming and calculi of discrete design, NATO ASI Series F36*, 1987.

[Bir89] R. Bird. Lectures on constructive functional programming. In M. Broy, editor, *Constructive Methods in Computer Science, NATO ASI Series F55*, 1989.

[Bis67] E. Bishop. *Foundations of Constructive Analysis*. McGraw-Hill, 1967.

[BKSS97] Y. Bertot, T. Kleymann-Schreiber, and D. Sequeira. Implementing proof by pointing without a structure editor. Technical Report ECS-LFCS-97-368, Dept of Computer Science, University of Edinburgh, 1997.

[BL84] R. Burstall and B. Lampson. Pebble, a kernel language for modules and abstract data types. *Lecture Notes in Computer Science*, 173, 1984.

[BL88] R. Burstall and Z. Luo. A set-theoretic setting for structuring theories in proof development. Circulated notes, April 1988.

[BL90a] K. Bruce and G. Longo. A modest model of records, inheritance, and bounded quantification. *Information and Computation*, 87, 1990.

[BL90b] R. Burstall and Z. Luo. A set-theoretic setting for structuring theories in proof development. manuscript, 1990.

[BM90] R. Burstall and J. McKinna. Deliverables : an approach to program development in the calculus of constructions. In the preliminary Proceedings of the 1st Workshop on Logical Frameworks, 1990.

[BM91] R. Burstall and J. McKinna. Deliverables : an approach to program development in the calculus of constructions. LFCS report ECS-LFCS-91-133, Dept of Computer Science, University of Edinburgh, 1991.

[BM92] R. Burstall and J. McKinna. Deliverables : a categorical approach to program development in type theory. LFCS report ECS-LFCS-92-242, Dept of Computer Science, University of Edinburgh, 1992.

[BMS80] R. Burstall, D. MacQueen, and D. Sannella. Hope : an experimental applicative language. In *1980 LISP Conference*, California, 1980.

[Bou96] R. Boulton. A tool to support formal reasoning about computer languages. Technical Report TR405, Computer Lab, Univ of Cambridge, 1996.

[Bry86] R. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. on Computers*, 35(8), 1986.

[BS92] J. Bradfield and C. Stirling. Local model checking for infinite state spaces. *Theoretical Computer Science*, 96, 1992.

[Bur84] R. Burstall. Programming with modules as typed functional programming. In *Proc. of Inter. Conf. on Fifth Generation Computer Systems*, Tokyo, 1984.

[Bur86] R. Burstall. Research in interactive theorem proving at edinburgh university. In *Proc. of 20th IBM Computer Science Symposium*, Shizuoka, Japan, 1986. Also, LFCS Report ECS-LFCS-86-12, Dept. of Computer Science, Univ. of Edinburgh.

[Bur89a] R. Burstall. An approach to program specification and development in constructions. Talk given in Workshop on Programming Logic, Bastad, Sweden, May 1989.

[Bur89b] R. Burstall. Computer-assisted proof for mathematics : an introduction, using the lego proof system. To appear in Proc. of the Institute for Applied Math. conf., Brighton Polytechnic, 1989.

[Bur93] R. Burstall. Extended Calculus of Constructions as a specification language. In R. Bird and C. Morgan, editors, *Mathematics of Program Construction*, 1993. Invited talk.

[BYBL95] T. Bull, E. Younger, K. Bennett, and Z. Luo. Bylands : reverse engineering safety-critical systems. *Proc. of Inter. Conf. on Software Maintenance*, 1995.

[C$^+$86] R.L. Constable et al. *Implementing Mathematics with the NuPRL Proof Development System*. Pretice-Hall, 1986.

[Car37] R. Carnap. *The Logical Syntax of Language*. ?, 1937.

[Car78] J. Cartmell. *Generalized Algebraic Theories and Contextual Category*. PhD thesis, University of Oxford, 1978.

[Car83] R. Carnap. The logicist foundations of mathematics. In P. Benacerraf and H. Putnam, editors, *Philosophy of Mathematics : selected readings*. Cambridge University Press, 2 edition, 1983.

[Car86a] L. Cardelli. A polymorphic λ-calculus with type :type. manuscript, 1986.

[Car86b] J. Cartmell. Generalized algebraic theories and contextual category. *Annals of Pure and Applied Logic*, 32, 1986.

[Car88] L. Cardelli. Type-checking dependent types and subtypes. *Lecture Notes in Computer Science*, 306, 1988.

[Car89] L. Cardelli. Typeful programming. Lecture notes for the IFIP State of the Art Seminar on Formal Description of Programming Concepts, Rio de Janeiro, Brazil, 1989.

[Car97] L. Cardelli. Program fragments, linking, and modularisation. *Proc. of POPL97*, 1997.

[Cas97] G. Castagna. *Object-Oriented Programming : a Unified Foundation*. Birkhäuser, 1997.

[CC91] P.-L. Curien and R. Di Cosmo. A confluent reduction for the -calculus with surjective pairing and terminal object. In *Proc. ICALP'91*, 1991.

[CF58] H.B. Curry and R. Feys. *Combinatory Logic*, volume 1. North Holland Publishing Company, 1958.

[CG90] Th. Coquand and J.H. Gallier. A proof of strong normalization for the theory of constructions using a Kripke-like interpretation. In *Preliminary Proc. of the Workshop on Logical Frameworks*, Antibes, 1990.

[CGL92] E. Clarke, O. Grunmberg, and D. Long. Model-checking and abstraction. *Proc of 19th Ann. ACM Symp on Principles of Programming Languages (POPL'92)*, 1992.

[CGW87] Th. Coquand, C. Gunter, and G. Winskel. Domain theoretic models of polymorphism. Tech. Report 116, Computer Laboratory, University of Cambridge, 1987.

[CH85] Th. Coquand and G. Huet. Constructions : a higher order proof system for mechanizing mathematics. *Lecture Notes in Computer Science*, 203, 1985.

[CH88] Th. Coquand and G. Huet. The calculus of constructions. *Information and Computation*, 76(2/3), 1988.

[Chu40] A. Church. A formulation of the simple theory of types. *J. Symbolic Logic*, 5(1), 1940.

[CL98] P. Callaghan and Z. Luo. Computer-assisted reasoning with natural language : implementing a mathematical vernacular. 1998. Draft paper submitted.

[Com95] A. Compagnoni. *Higher-order Subtyping with Intersection Types*. PhD thesis, Katholieke Universiteit Nijmegen, 1995.

[Con71] R.L. Constable. Constructive mathematics and automatic programs writers. In *Proc. IFIP'71*, 1971.

[Coq85] Th. Coquand. *Une Theorie des Constructions*. PhD thesis, University of Paris VII, 1985.

[Coq86a] Th. Coquand. An analysis of Girard's paradox. In *Proc. 1st Ann. Symp. on Logic in Computer Science*. IEEE, 1986.

[Coq86b] Th. Coquand. A calculus of constructions. manuscript, 1986.

[Coq89] Th. Coquand. Metamathematical investigations of a calculus of constructions. manuscript, 1989.

[Coq91] Th. Coquand. An algorithm for testing conversion in Type Theory. In G. Huet and G. Plotkin, editors, *Logical Frameworks*. Cambridge University Press, 1991.

[Coq92] Th. Coquand. Pattern matching with dependent types. Talk given at the BRA workshop on Proofs and Types, Bastad, 1992.

[Coq96] Coq. *The Coq Proof Assistant Reference Manual (version 6.1)*. INRIA-Rocquencourt and CNRS-ENS Lyon, 1996.

[Cou97] J. Courant. A module system for pure type systems. *Proc of TL-CA'97*, 1997.

[CPM90] Th. Coquand and Ch. Paulin-Mohring. Inductively defined types. *Lecture Notes in Computer Science*, 417, 1990.

[Cro65] Dummett Crossley. Formal systems and recursive fonctions. In *Logic colloquium*, Amsterdam, 1965. North-Holland.

[Cur95] P. Curzon. The importance of proof maintenance and reengineering. 1995.

[CW85] L. Cardelli and P. Wegner. On understanding types, data abstraction and polymorphism. *Computing Surveys*, 17, 1985.

[D^+91] G. Dowek et al. *The Coq Proof Assistent : User's Guide (version 5.6)*. INRIA-Rocquencourt and CNRS-ENS Lyon, 1991.

[dB72] N.G. de Bruijn. Lambda calculus notation with nameless dummies : a tool for automatic formula manipulation with application to the church-rosser theorem. Indag. Mathematics 34, 1972.

[dB80] N.G. de Bruijn. A survey of the project AUTOMATH. In J. Hindley and J. Seldin, editors, *To H. B. Curry : Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, 1980.

[dB94] N. G. de Bruijn. The mathematical vernacular, a language for mathematics with typed sets. In R. P. Nederpelt, J. H. Geuvers, and R. C. de Vrijer, editors, *Selected Papers on Automath*. North Holland, 1994.

[Dev79] K. Devlin. *Fundamentals of Contemporary Set Theory*. Springer-Verlag, 1979.

[Dij68] E.W. Dijkstra. Cooperating sequential processes. In F. Genuys, editor, *Programming Languages*. Academic Press, 1968.

[Dow93] G. Dowek. The undecidability of typability in the lambda-Pi calculus. *Proc of Typed Lambda Calculi and Applications, LNCS 664*, 1993.

[DP96] R. Dyckhoff and L. Pinto. A permutation-free sequent calculus for intuitionistic logic. Technical Report CS/96/9, Computer Science Division, St Andrews University, 1996.

[Dum75a] M. Dummett. The philosophical basis of intuitionistic logic. In H. Rose and J. Shepherdson, editors, *Proc. of the Logic Colloquium, 1973*. North Holland, 1975. Reprinted in P. Benacerraf and H. Putnam (eds.), Philosophy of Mathematics : selected readings, Cambridge University Press.

[Dum75b] M. Dummett. The philosophical foundations of intuitionistic logic. In H.Rose and J.C.Shepherdson, editors, *Logic Colloquium'73*, pages 5–40, 1975.

[Dum91] M. Dummett. *The Logical Basis of Metaphysics*. Duckworth, 1991.

[Dyb88] P. Dybjer. Inductively defined sets in 's set theory. In A. Avron et al, editor, *Workshop on General Logic*. LFCS Report Series, ECS-LFCS-88-52, Dept. of Computer Science, University of Edinburgh, 1988.

[Dyb89] P. Dybjer. An inversion principle for 's type theory. In P. Dybjer et al, editor, *Workshop on Programming Logic*. Programming Methodology Group, Report 54, 1989.

[Dyb91] P. Dybjer. Inductive sets and families in 's type theory and their set-theoretic semantics. In G. Huet and G. Plotkin, editors, *Logical Frameworks*. Cambridge University Press, 1991.

[Dyb93] Peter Dybjer. Inductive families. *Formal Aspects of Computing*, 1993. To appear.

[Dyb94] P. Dybjer. A general formulation of simultaneous inductive-recursive definitions in type theory. 1994.

[EFH83] H. Ehrig, W. Fey, and H. Hansen. ACT ONE : an algebraic specification language with two levels of semantics. Technical Report 83-03, Technical University of Berlin, Fachbereich Informatik, 1983.

[Ehr88] T. Ehrhard. A categorical semantics of constructions. *Proc. 3rd Ann. Symp. on Logic in Computer Science, Edinburgh. IEEE*, 1988.

[EKMP82] H. Ehrig, H. Kreowski, B. Mahr, and P. Padawitz. Algebraic implementation of abstract data types. *Theoretical Computer Science*, 20 :209–263, 1982.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1 : Equations and Initial Semantics*. Springer, 1985.

[EM90] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 2 : Module specifications and Constraints*. Springer, 1990.

[FGJM85] K. Futatsugi, J. Goguen, J.-P. Jouannaud, and J. Meseguer. Principles of OBJ2. *Proc. Principles of Programming Languages. ACM*, 1985.

[FH94] A. Felty and D. Howe. Generalisation and reuse of tectic proofs. *Proc. of 5th Inter. Conf. on Logic Programming and Automated Reasoning*, 1994.

[FL97] M. Fox and D. Long. Formalising non-linear planning. Manuscript submitted for publication, 1997.

[Fri77] H. Friedman. Set-theoretic foundations for constructive analysis. *Annals of Mathematics*, 105, 1977.

[Gal90] J.H. Gallier. On Girard's 'Candidats de Reductibilité'. In P. Odifreddi, editor, *Logic and Computer Science*. Academic Press, 1990.

[GB80] J. Goguen and R. Burstall. Cat : a system for the structured elaboration of correct programs and from structured specifications. Tech Report CSL-118, SRI International, 1980.

[GB84] J. Goguen and R. Burstall. Introducing institutions. *LNCS 164*, 1984.

[GB90] J. Goguen and R. Burstall. Institutions : abstract model theory for specification and programming. Technical Report ECS-LFCS-90-106, LFCS, Dept of Computer Science, University of Edinburgh, 1990.

[Gen35] G. Gentzen. Untersuchungen über das logische schliessen. *Mathematische Zeitschrift*, 39, 1935.

[GHM76] J.V. Guttag, E. Horowitz, and D.R. Musser. Abstract data types and software validation. *Comm. ACM*, 21(12), 1976.

[Gir71] J.-Y. Girard. Une extension de l'interpretation fonctionelle de gödel à l'analyse et son application à l'élimination des coupures dans et la thèorie des types'. *Proc. 2nd Scandinavian Logic Symposium.* North-Holland, 1971.

[Gir72] J.-Y. Girard. *Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur*. PhD thesis, Université Paris VII, 1972.

[Gir73] J.-Y. Girard. Quelques résultats sur les interprétations fonctionells. *Lecture Notes in Mathematics 337.* Springer, 1973.

[Gir86] J.-Y. Girard. The system F of variable types, fifteen years later. *Theoretical Computer Science*, 45, 1986.

[GL91] H. Goguen and Z. Luo. Inductive data types : Well-ordering types revisited. submitted manuscript, 1991.

[GL92] H. Goguen and Z. Luo. Inductive data types : Well-ordering types revisited. In G. Huet and G. Plotkin, editors, *Logical Frameworks*. Cambridge University Press, 1992. (To appear.) Also as LFCS Report Series ECS-LFCS-92-209, Dept of Computer Science, University of Edinburgh.

[GL93a] H. Goguen and Z. Luo. Inductive data types : Well-ordering types revisited. In G. Huet and G. Plotkin, editors, *Logical Environments*. Cambridge University Press, 1993. Also as LFCS Report Series ECS-LFCS-92-209, Dept of Computer Science, University of Edinburgh.

[GL93b] H. Goguen and Z. Luo. Inductive data types : Well-ordering types revisited. In G. Huet and G. Plotkin, editors, *Logical Environments*. Cambridge University Press, 1993.

[GLT90] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge University Press, 1990.

[GM93] M. Gordon and T. Melham. *Introduction to HOL : a theorem proving environment for higher-order logic*. Cambridge University Press, 1993.

[GMW79] M. Gordon, R. Milner, and C. Wadsworth. *LCF : a mechanised logic of computation*, volume 78 of *LNCS*. Springer-Verlag, 1979.

[Gog94] H. Goguen. *A Typed Operational Semantics for Type Theory*. PhD thesis, University of Edinburgh, 1994.

[Gog95] H. Goguen. Typed operational semantics. *Proceedings of the 2nd Inter. Conf. on Typed Lambda Calculi and Applications, Edinburgh*, 1995.

[GTW78] J.A. Goguen, J.W. Thatcher, and E.G. Wagner. Abstract data types as initial algebras and the correctness of data representation. In R. Yeh, editor, *Current Trends in Programming Methodology, Vol. 4*. Prentice Hall, 1978.

[Har95] J. Harrison. Binary decision diagrams as a hol derived rule. *The Computer Journal*, 38(1), 1995.

[Hay89] S. Hayashi. Constructive mathematics and computer-assisted reasoning systems. In *Proc. of Heyting'88*, 1989.

[Hay91] S. Hayashi. Singleton, union and intersection types in program extraction. *Lecture Notes in Computer Science*, 526, 1991. Revised version to appear in Information and Computation.

[Hey71] A. Heyting. *Intuitionism : An Introduction*. North-Holland, 1971.

[HH86] J. Hook and D. Howe. Impredicative strong existential equivalent to Type :Type. Technical Report TR86-760, Cornell University, 1986.

[HHP87] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *Proc. 2nd Ann. Symp. on Logic in Computer Science. IEEE*, 1987.

[Hoa72] C.A.R. Hoare. Proofs of correctness of data representation. *Acta Informatica*, 1(1), 1972.

[Hof92a] M. Hofmann. Formal development of functional programs in type theory. LFCS report ECS-LFCS-92-228, Dept. of Computer Science, Edinburgh University, 1992. submitted to Science of Computer Programming.

[Hof92b] M. Hofmann. Formal development of functional programs in type theory. LFCS report ECS-LFCS-92-228, Dept. of Computer Science, Edinburgh University, 1992. submitted to Science of Computer Programming.

[Hof93a] Martin Hofmann. Elimination of extensionality in Martin-Löf type theory. To appear in the proceedings of the workshop on types for proofs and programs, Nijmegen, June 1993.

[Hof93b] Martin Hofmann. Sound and complete axiomatisations for call-by-value control operators. In *Proceedings of the 5th conference on category theory in Computer Science CTCS, Amsterdam*, 1993. To appear as a special issue of MSCS.

[Hof95] M. Hofmann. *Extensional concepts in intensional type theory*. PhD thesis, Department of Computer Science, University of Edinburgh, 1995.

[How80] W. A. Howard. The formulae-as-types notion of construction. In J. Hindley and J. Seldin, editors, *To H. B. Curry : Essays on Combinatory Logic*. Academic Press, 1980.

[HP87] M. Hyland and A. Pitts. The theory of constructions : categorical semantics and topos-theoretic models. *Categories in Computer Science and Logic, Boulder*, 1987.

[HP89] R. Harper and R. Pollack. Type checking, universe polymorphism, and typical ambiguity in the calculus of constructions. *Theoretical Computer Science*, 1989. to appear.

[HP91] R. Harper and R. Pollack. Type checking, universe polymorphism, and typical ambiguity in the calculus of constructions. *Theoretical Computer Science*, 89(1), 1991.

[HP92] Martin Hofmann and Benjamin Pierce. An abstract view of objects and subtyping (preliminary report). Technical Report ECS-LFCS-92-226, University of Edinburgh, LFCS, 1992. submitted to STACS '94, Caen, France.

[HS87] J.R. Hindley and J.P. Seldin. *Introduction to Combinators and λ-calculus*. Cambridge University Press, 1987.

[HS93] Martin Hofmann and Thomas Streicher. A model of intensional Martin-Löf type theory in which unicity of identity proofs does not hold. To be submitted to LICS '94, July 1993.

[HS96] K. Haveland and N. Shankar. Experiments in theorem proving and model checking for protocol verification. *Proc Formal Methods Europe FME'96*, 1996.

[HSV94] L. Helmink, M. Sellink, and F. Vaandrager. Proof-checking a data link protocol. *TYPES'94*, 1994.

[Hue87a] G. Huet. A calculus with type :type. unpublished manuscript, 1987.

[Hue87b] G. Huet. Induction principles formalized in the calculus of constructions. *TAPSOFT'87, LNCS'249*, 1987.

[Hue89] G. Huet. (ed.) the calculus of constructions : Documentation and user's guide. Technical Report INRIA 110, INRIA, 1989.

[Hyl82] M. Hyland. The effective topos. In A.S. Troelstra and van Dalen, editors, *The Brouwer Symposium*. North-Holland, 1982.

[Hyl88] M. Hyland. A small complete category. *Annals of Pure and Applied Logic*, 40, 1988.

[Jac94] D. Jackson. Abstract model-checking of infinite specifications. *Proc of Formal Methods Europe (FME'94)*, 1994.

[Jac97] R. Jackendoff. *The Architecture of the Language Faculty*. MIT, 1997.

[Jay90] B. Jay. personal communication, 1990.

[JLS98] A. Jones, Z. Luo, and S. Soloviev. Some proof-theoretic and algorithmic aspects of coercive subtyping. *Proc. of the Annual Conf on Types and Proofs (TYPES'96)*, 1998.

[JM94] C. Jones and S. Maharaj. The Lego library. Technical report, LFCS, Department of Computer Science, University of Edinburgh, 1994.

[Jon86] C.B. Jones. *Systematic Software Development using VDM*. Prentice-Hall, 1986.

[Jon93] C. Jones. Completing the rationals and metric spaces in LEGO. In G. Huet and G. Plotkin, editors, *Logical Environments*. Cambridge University Press, 1993.

[Jon95] A. Jones. The formalization of linear algebra in lego : The decidable dependency theorem. Master's thesis, University of Manchester, 1995.

[KBY92] T. Bull K. Bennett and H. Yang. A transformation system for maintenance—turning theory into practice. *Proc. of Inter. Conf. on Software Maintenance, Florida*, 1992.

[Klo80] J. W. Klop. Combinatory reduction systems. *Mathematical Center Tracts 127*, 1980.

[Kol32] A.N. Kolmogorov. Zur deutung der intuitionistischen logik. *Math. Z.*, 35, 1932.

[Kre68] G. Kreisel. Functions, ordinals, species. In B. van Rootselaar and J. Staal, editors, *Logic, Methodology and Philosophy of Science III*, Amsterdam, 1968. North-Holland.

[LB88] B. Lampson and R. Burstall. Pebble, a kernel language for modules and abstract data types. *Information and Computation*, 76(2/3), 1988.

[LB92] Z. Luo and R. Burstall. A set-theoretic setting for structuring theories in proof development. LFCS Report Series ECS-LFCS-92-206, Department of Computer Science, University of Edinburgh, 1992.

[LC97] Z. Luo and P. Callaghan. Linguistic categories in mathematical vernacular and their type-theoretic semantics (extended abstract). *Logical Aspects of Computational Linguistics 97*, 1997.

[LC98a] Z. Luo and P. Callaghan. Coercive subtyping and lexical semantics. Submitted to LACL'98, 1998.

[LC98b] Z. Luo and P. Callaghan. Mathematical vernacular and conceptual well-formedness in mathematical language. *Proc of LACL97.*, 1998. To appear in LNCS series.

[Lev79] A. Levy. *Basic Set Theory*. Springer-Verlag, 1979.

[LG94] D. Long and R. Garigliano. *? ? ?* ? ? ?, 1994.

[LM88a] G. Longo and E. Moggi. Constructive natural deduction and its 'modest' interpretation. Report CMU-CS-88-131, Computer Science Dept., Carnegie Mellon Univ., 1988.

[LM88b] G. Longo and E. Moggi. Constructive natural deduction and its 'modest' interpretation. Technical Report CMU-CS-88-131, Computer Science Dept., Carnegie Mellon Univ., 1988.

[LMS95] G. Longo, K. Milsted, and S. Soloviev. A logic of subtyping. In *Proc. of LICS'95*, 1995.

[Lon88] G. Longo. Some aspects of impredicativity. Technical report CMU-CS-88-135, Department of Computer Science, CMU, 1988.

[LP92] Z. Luo and R. Pollack. LEGO Proof Development System : User's Manual. LFCS Report ECS-LFCS-92-211, Department of Computer Science, University of Edinburgh, 1992.

[LPT89] Z. Luo, R. Pollack, and P. Taylor. *How to Use LEGO : a preliminary user's manual*. LFCS Technical Notes LFCS-TN-27, Dept. of Computer Science, Edinburgh University, 1989.

[LS86] J. Lambek and P.J. Scott. *Introduction to Higher-Order Categorical Logic*, volume VII of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[LS97] Z. Luo and S. Soloviev, editors. *Proc of TYPES Working Group Workshop on Subtyping, Inheritance, and Modularisation of Proofs*, Durham, 1997.

[Luo88a] Z. Luo. $CC^{\infty}_{\subset}$ and its meta theory. LFCS report ECS-LFCS-88-58, Dept. of Computer Science, Edinburgh University, 1988.

[Luo88b] Z. Luo. A higher-order calculus and theory abstraction. LFCS Report ECS-LFCS-88-57, Dept. of Computer Science, Edinburgh University, 1988.

[Luo89a] Z. Luo. **ECC**, an extended calculus of constructions. In *Proc. of IEEE Fourth Ann. Symp. on Logic in Computer Science*, Asilomar, California, U.S.A., June 1989.

[Luo89b] Z. Luo. **ECC**, an extended calculus of constructions. In *Proc. of LICS'89*, 1989.

[Luo89c] Z. Luo. A higher-order calculus and theory abstraction. To appear in Information and Computation, 1989.

[Luo90a] Z. Luo. *An Extended Calculus of Constructions*. PhD thesis, University of Edinburgh, 1990. Also as Report CST-65-90/ECS-LFCS-90-118, Department of Computer Science, University of Edinburgh.

[Luo90b] Z. Luo. *An Extended Calculus of Constructions*. PhD thesis, Univ of Edinburgh, 1990.

[Luo90c] Z. Luo. A problem of adequacy : conservativity of calculus of constructions over higher-order logic. Technical report, LFCS report series ECS-LFCS-90-121, Department of Computer Science, University of Edinburgh, 1990.

[Luo90d] Z. Luo. A unifying theory of dependent types. manuscript., 1990.

[Luo90e] Z. Luo. A unifying theory of dependent types I. manuscript, November 1990.

[Luo90f] Z. Luo. A unifying theory of dependent types I. manuscript, December 1990.

[Luo91a] Z. Luo. A higher-order calculus and theory abstraction. *Information and Computation*, 90(1), 1991.

[Luo91b] Z. Luo. Program specification and data refinement in type theory. *Proc. of the Fourth Inter. Joint Conf. on the Theory and Practice of Software Development (TAPSOFT), LNCS 493*, 1991. Also as LFCS report ECS-LFCS-91-131, Dept. of Computer Science, Edinburgh University.

[Luo91c] Z. Luo. Type theory, logic and computer science. PG course notes, January 1991.

[Luo91d] Z. Luo. A unifying theory of dependent types I. Technical report, LFCS report series ECS-LFCS-91-154, 1991.

[Luo91e] Z. Luo. A unifying theory of dependent types I. manuscript, April 1991.

[Luo92a] Z. Luo. A unifying theory of dependent types : the schematic approach. *Proc. of Symp. on Logical Foundations of Computer Science (Logic at Tver'92), LNCS 620*, 1992. Also as LFCS Report ECS-LFCS-92-202, Dept. of Computer Science, University of Edinburgh.

[Luo92b] Z. Luo. A unifying theory of dependent types : the schematic approach. *Proc. of Symp. on Logical Foundations of Computer Science (Logic at Tver'92), LNCS 620*, 1992.

[Luo93a] Z. Luo. Program specification and data refinement in type theory. *Mathematical Structures in Computer Science*, 3(3), 1993. An earlier version appears as [Luo91b].

[Luo93b] Z. Luo. Program specification and data refinement in type theory. *Mathematical Structures in Computer Science*, 3(3), 1993.

[Luo94a] Z. Luo. Algorithm refinement in type theory. manuscript, September 1994.

[Luo94b] Z. Luo. *Computation and Reasoning : A Type Theory for Computer Science*. Oxford University Press, 1994.

[Luo94c] Z. Luo. Generic programming and program transformation. In preparation, 1994.

[Luo95a] Z. Luo. Developing reuse technology in proof engineering. *Proceedings of AISB95, Workshop on Automated Reasoning : bridging the gap between theory and practice, Sheffield, U.K.*, April 1995.

[Luo95b] Z. Luo. Formal verification of concurrent algorithms in type theory. 1995. Submitted to TACAS96 for publication.

[Luo95c] Z. Luo. Formalisation of concurrent programs in type theory. Lecture notes for the Lego Summer School, Germany, 1995.

[Luo96] Z. Luo. Logical truths in constructive type theory (abstract). *Logic Colloquium 96*, 1996.

[Luo97a] Z. Luo. Coercive subtyping. 1997. Draft.

[Luo97b] Z. Luo. Coercive subtyping. *Journal of Logic and Computation*, 1997. Accepted, to appear in 1998.

[Luo97c] Z. Luo. Coercive subtyping in type theory. *Proc. of CSL'96, the 1996 Annual Conference of the European Association for Computer Science Logic, Utrecht. LNCS 1258*, 1997.

[Luo97d] Z. Luo. Mathematical vernacular. Invited talk at the ESSILLI Workshop on Informal Mathematical Languages, 1997.

[Luo98] Z. Luo. Coercive subtyping. *Journal of Logic and Computation*, 1998. To appear.

[LY95] Z. Luo and S. Yu. Model-checking as a proof procedure in type theory. 1995. Draft.

[Lyo95] J. Lyons. *Linguistic Semantics*. Cambridge University Press, 1995.

[LZ75] B. Liskov and S. Zilles. Specification techniques for data abstraction. *IEEE Trans. on Software Engineering*, SE-1(1), 1975.

[Mac81] D.D. MacQueen. Structures and parameterization in a typed functional language. *Proc. Symp. on Functional Programming and Computer Architecture*, 1981.

[Mac86] D. MacQueen. Using dependent types to express modular structure. *Proc. 13th Principles of Programming Languages*, 1986.

[Mag92] L. Magnusson. The new implementation of ALF. *Informal Proceedings of Workshop on Logical Frameworks, Bastad*, 1992.

[McK92] J. McKinna. *Deliverables : a categorical approach to program development in type theory*. PhD thesis, Department of Computer Science, University of Edinburgh, 1992.

[Mee86] L. Meertens. Algorithmics : towards programming as a mathematical activity. In *Proc. CWI Symp. on Mathematics and Computer Science*, 1986.

[Men87] N. Mendler. Recursive types and type constraints in second-order lambda calculus. In *Proceedings of POPL*, 1987.

[Men88] N. Mendler. *Recursive Definition in Type Theory*. PhD thesis, Cornell University, 1988.

[Men93] M. Mendler. *A Modal Logic for Handling Behavioural Constraints in Formal Hardware Verification*. PhD thesis, University of Edinburgh, 1993. Also as Report ECS-LFCS-93-255/CST-97-93, Dept. of Computer Science, Edinburgh University.

[Mes88] J. Meseguer. Relating models of polymorphism. Technical Report SRI-CSL-88-13, Computer Science Lab, SRI International, 1988.

[Mes89] J. Meseguer. General logics. In H.-D. Ebbinghaus et al, editor, *Logic Colloquium 1987*. North-Holland, 1989.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[Miq01] Alexandre Miquel. *Le calcul des constructions implicite : syntaxe et sémantique*. PhD thesis, Université Paris 7, 2001.

[Mit86] J.C. Mitchell. A type inference approach to reduction properties and semantics of polymorphic expressions. *Proc. 1986 ACM Symp. on Lisp and Functional Programming*, 1986.

[ML71] P. Martin-Löf. An intuitionistic theory of types. manuscript, 1971.

[ML72] P. Martin-Löf. An intuitionistic theory of types. manuscript, 1972.

[ML75] P. Martin-Löf. An intuitionistic theory of types : predicative part. In H.Rose and J.C.Shepherdson, editors, *Logic Colloquium'73*, 1975.

[ML82] P. Martin-Löf. Constructive mathematics and computer programming. In L.J. Cohen , editor, *Logic, Methodology and Philosophy of Science VI*, Amsterdam, 1982. North-Holland.

[ML84] P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.

[ML85] P. Martin-Löf. Truth of a proposition, evidence of a judgement, validity of a proof. Talk given at Workshop of Theories of Meaning, Florence, June 1985.

[ML90] P. Martin-Löf. Mathematics of infinity. *COLOG'88, Lecture Notes in Computer Science 417*, 1990.

[ML94] P. Martin-Löf. Analytic and synthetic judgements in type theory. In P. Parrini, editor, *Kant and Contemporary Epistemology*. Kluwer Accademic Publishers, 1994.

[MM96] John C. Mitchell and Eugenio Moggi. Kripke-style models for typed lambda calculus. *Annals of Pure and Applied Logic*, 51 :99–124, 1996.

[MMS07] Lionel Marie-Magdeleine and Sergei Soloviev. Non-standard reductions in simply typed, higher order and dependently-typed systems. In HOR2007, editor, *Talk*, 2007.

[MN94] L. Magnusson and B. Nordström. The ALF proof editor and its proof engine. In *Types for Proof and Programs, LNCS*, 1994.

[Mog85] E. Moggi. The PER-model as internal category with all small products. manuscript, 1985.

[Mor94] G. V. Morrill. *Type Logical Grammar*. Kluwer Academic Publishers, 1994.

[MP81] Z. Manna and A. Pnueli. Verification of concurrent programs (part I, II, III). Technical report, Stanford University, 1981.

[MP85] J. Mitchell and G. Plotkin. Abstract types have existential type. *Proc. 12th Principles of Programming Languages*, 1985.

[MP93] J. McKinna and R. Pollack. Pure type systems formalized. In M.Bezem and J.F.Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications, TLCA'93*, pages 289–305. Springer-Verlag, LNCS 664, March 1993.

[MP95] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems : Safety*. Sringer-Verlag, 1995.

[MSV83] T.S.E. Maibaum, M.R. Sadler, and P.A.S. Veloso. Logical implementation, 1983.

[MTH90] R. Milner, M. Tofte, and R. Harper. *The Definition of Standard ML*. MIT, 1990.

[Myh75] J. Myhill. Constructive set theory. *J. Symbolic Logic*, 40, 1975.

[NP83] B. Nordström and K. Petersson. Types and specifications. *Proceedings of IFIP'83*, pages 915–920, 1983.

[NPS90] B. Nordström, K. Petersson, and J. Smith. *Programming in 's Type Theory : An Introduction*. Oxford University Press, 1990.

[NPS94] B. Nordström, K. Petersson, and J. Smith. 's type theory. Manuscript, 1994.

[Ore90] C.-E. Ore. The Extended Calculus of Constructions (**ECC**) with inductive types. To appear in Information and Computation, 1990.

[Ore92] C.-E. Ore. The Extended Calculus of Constructions (**ECC**) with inductive types. *Information and Computation*, 99(2) :231–264, 1992.

[PA93] G. Plotkin and M. Abadi. A logic for parametric polymorphism. In M. Bezem and J. Groote, editors, *Typed Lambda Calculi and Applications, LNCS 664*, 1993.

[PAC94] G. Plotkin, M. Abadi, and L. Cardelli. Subtyping and parametricity. 1994.

[Pau87] L. Paulson. *Logic and Computation : Interactive Proof with Cambridge LCF*. Cambridge University Press, 1987.

[Pau91] L. Paulson. *ML for the Working Programmer*. Cambridge University Press, 1991.

[Pau93] L. Paulson. Introduction to isabelle. Technical Report 280, Computer Laboratory, Cambridge University, 1993.

[Pet81] G. L. Peterson. Myths about the mutual exclusion problem. *Information Processing Letters*, 12(3), 1981.

[Pfe93] F. Pfenning. Refinement types for logical frameworks. *Preliminary Proceedings of BRA Workshop on Types and Proofs*, 1993.

[Pit87] A. Pitts. Polymorphism is set theoretic, constructively. *Proc. of the Summer Conf. on Category Theory and Computer Science*, 1987. Edinburgh.

[Plo81] G. Plotkin. A structural approach to operational semantics. Technical Report DAIMI-FN-19, Computer Science Dept, Arhus University, 1981.

[PM89] Ch. Paulin-Mohring. Extracting F programs from proofs in the calculus of constructions. *Proc. Principles of Programming Languages*, 1989.

[PM93] C. Paulin-Mohring. Inductive definitions in the system Coq : rules and properties. *Proceedings of the Inter. Conf. on Typed Lambda Calculi and Applications (TLCA'93), LNCS 664*, 1993.

[Pol] R. Pollack. *Theory of LEGO*. PhD thesis, Edinburgh University. forthcoming.

[Pol89] R. Pollack. The theory of LEGO. manuscript, 1989.

[Pol90a] R. Pollack. Implicit syntax. In the preliminary Proceedings of the 1st Workshop on Logical Frameworks, 1990.

[Pol90b] R. Pollack. The Tarski fixpoint theorem. communication on TYPES e-mail network, 1990.

[Pol92] R. Pollack. Typechecking in pure type systems. In *Informal Proceedings of the 1992 Workshop on Types for Proofs and Programs, Båstad, Sweden*, pages 271–288, June 1992. available by ftp.

[Pol94a] R. Pollack. *The Theory of LEGO : A Proof Checker for the Extended Calculus of Constructions*. PhD thesis, Edinburgh University, 1994.

[Pol94b] R. Pollack. *The Theory of LEGO : a proof checker for the Extended Calculus of Constructions*. PhD thesis, Edinburgh University, 1994.

[Pol95a] Email communications, 1995.

[Pol95b] R. Pollack. Are tactics feasible ? *2nd Conf. on Typed Lambda Calculi*, 1995.

[Pol95c] R. Pollack. A verified proof checker. *Proceedings of the 2nd Inter. Conf. on Typed Lambda Calculi and Applications, Edinburgh*, April 1995.

[Pol97a] R. Pollack. How to believe a machine-checked proof,. In G. Sambin and J. Smith, editors, *Twenty-Five Years of Constructive Type Theory*. Oxford University Press, 1997.

[Pol97b] R. Pollack. Theories in type theory. *In [LS97]*, 1997.

[Pot87] G. Pottinger. Strong normalization for terms of the theory of constructions. Technical Report TR 11-7, Odyssey Research Associates, 1987.

[Pra65] D. Prawitz. *Natural Deduction, a Proof-Theoretic Study*. Lmqvist and Wiksell, 1965.

[Pra73] D. Prawitz. Towards a foundation of a general proof theory. In P. Suppes , editor, *Logic, Methodology, and Phylosophy of Science IV*, 1973.

[Pra74] D. Prawitz. On the idea of a general proof theory. *Synthese*, 27, 1974.

[Pus95] J. Pustejovsky. *The Generative Lexicon*. MIT, 1995.

[Qui62] W.V. Quine. Carnap and logical truth. In B. Kazemier and D. Vuysje, editors, *Logic and Language : Studies Dedicated to Professor Rudolf Carnap*, 1962.

[Qui86] W.V. Quine. *Philosophy of Logic*. Harvard University Press, second edition, 1986.

[Ram25] F.P. Ramsey. The foundations of mathematics. *Proceedings of the London Mathematical Society*, 25 :338–384, 1925.

[Ran94] A. Ranta. *Type-theoretical Grammar*. Oxford University Press, 1994.

[Ran95a] A. Ranta. Syntactic categories in the language of mathematics. *LNCS 996*, 1995.

[Ran95b] A. Ranta. Type-theoretical interpretation and generalization of phrase structure grammar. *Bulletin of the IGPL*, 1995.

[Rey74] J.C. Reynolds. Towards a theory of type structure. *Lecture Notes in Computer Science*, 19, 1974.

[Rey83] J. C. Reynolds. Types, abstraction and parameter polymorphism. *Information Processing'83*, 1983.

[Rey84] J. C. Reynolds. Polymorphism is not set-theoretic. *Lecture Notes in Computer Science*, 173, 1984.

[RFMMS09] Maxime Rebout, Louis Féraud, Lionel Marie-Magdeleine, and Sergei Soloviev. Computations in Graph Rewriting : Inductive types and Pullbacks in DPO Approach (regular paper). In Tomasz Szmuc, Marcin Szpyrka, and Jaroslav Zendulka, editors, *IFIP TC2 Central and East European Conference on Software Engineering Techniques - CEE-SET, Krakow, Pologne, 12/10/09-14/10/09*, pages 164–177, http ://www.springerlink.com/, 2009. Springer-Verlag.

[Rit87] B. Ritchie. *The Design and Implementation of an Interactive Proof Editor*. PhD thesis, University of Edinburgh, 1987.

[RP88] J. C. Reynolds and G. D. Plotkin. On functors expressible in the polymorphic typed lambda calculus. LFCS report ECS-LFCS-88-53, Dept. of Computer Science, Univ. of Edinburgh, 1988.

[RS92] B. Reus and Th. Streicher. Lifting the laws of module algebra to deliverables. Technical report, Ludwig-Maximilians-Universität Münichen, Institut für Informatik, 1992.

[Rus03] B.A.W. Russell. *The Principles of Mathematics*. Routledge, 1903. Paperback edition, 1992.

[Sai96] A. Saibi. Typing algorithm in type theory with inheritance. Draft, 1996.

[Sai97] A. Saibi. Typing algorithm in type theory with inheritance. *Proc of POPL'97*, 1997.

[Sal89] A. Salvesen. The church-rosser theorem for lf with $\beta\eta$ reduction. manuscript, 1989.

[SB83] D. Sannella and R. Burstall. Structured theories in LCF. *Proc. of the 8th Colloquium on Trees in Algebra and Programming*, 1983.

[SC03] Sergei Soloviev and David Chemouil. Some Algebraic Structures in Lambda-Calculus with Inductive Types . In Stefano Berardi, Mario Coppo, and Ferrucio Damiani, editors, *TYPES , Torino, 30/04/03-04/05/03*, pages 338–354. Springer, LNCS, Volume 3085/2004, avril 2003.

[Sch97] Thomas Schreiber. Auxiliary variables and recursive procedures. *TAPSOFT'97 : Theory and Practice of Software Development, LNCS 1214*, 1997.

[Sco70] D. Scott. Constructive validity. *Symp. on Automatic Demonstration, Lecture Notes in Mathematics 125*, 1970.

[See86] R.A.G. Seely. Categorical semantics for higher-order polymorphic lambda calculus. *J. of Symbolic Logic*, 52(4), 1986.

[SL98a] S. Soloviev and Z. Luo. Coercive subtyping : coherence and conservativity. Talk given in and paper submitted to TYPES'98, 1998.

[SL98b] S. Soloviev and Z. Luo. Meta-theoretic study of coercive subtyping. Technical report, CS Dept, Durham, 1998.

[SL00] Sergei Soloviev and Zhaohui Luo. Coercion completion and conservativity in coercive subtyping. In *ANNALS OF PURE AND APPLIED LOGIC*, pages 113–1. Addison-Wesley, 2000.

[SLG96] S. Shiu, Z. Luo, and R. Garigliano. Type-theoretic semantics for semnet. *Proc. of Inter. Conf. on Formal and Applied Practical Reasoning, LNAI 1085*, 1996.

[Smi88] J. Smith. The independence of Peano's fourth axiom from 's type theory without universes. *Journal of Symbolic Logic*, 53(3), 1988.

[SMMS91] Luca Cardelli Simone, Simone Martini, John C. Mitchell, and Andre Scedrov. An extension of system f with subtyping, 1991.

[SP94] Martin Steffen and Benjamin Pierce. Higher-order subtyping, 1994.

[SS88] A. Salvesen and J. Smith. The strength of the subset type in 's type theory. *LICS'88*, 1988.

[SST92a] D. Sannella, S. Sokolowski, and A. Tarlecki. Toward formal development of programs from algebraic specifications : Parameterization revisited. LFCS Report Series ECS-LFCS-92-222, Dept. of Computer Science, University of Edinburgh, 1992.

[SST92b] D. Sannella, S. Sokolowski, and A. Tarlecki. Toward formal development of programs from algebraic specifications : Parameterization revisited. LFCS Report ECS-LFCS-92-222, Department of Computer Science, University of Edinburgh, 1992.

[ST87a] D. Sannella and A. Tarlecki. Extended ML : an institution-independent framework for formal program development. *Proc. Workshop on Category Theory and Computer Programming, LNCS 240*, pages 364–389, 1987.

[ST87b] D. Sannella and A. Tarlecki. On observational equivalence and algebraic specification. *Computer and System Science*, 34, 1987.

[ST88a] D. Sannella and A. Tarlecki. Specifications in arbitrary institutions. *Information and Computation*, 76, 1988.

[ST88b] D. Sannella and A. Tarlecki. Toward formal development of programs from algebraic specifications : implementation revisited. *Acta Informatica*, 25, 1988.

[ST90] D. Sannella and A. Tarlecki. A kernel specification formalism with higher-order parameterization. Draft, 1990.

[Sti92] C. Stirling. Modal and temporal logics. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*. Oxford University Press, 1992.

[Sti95] C. Stirling. Modal and temporal logics for processes. 1995.

[Str88] T. Streicher. *Correctness and Completeness of a Categorical Semantics of the Calculus of Constructions*. PhD thesis, Passau, 1988.

[Str91] T. Streicher. private communication, 1991.

[Str93] T. Streicher. Investigations into intensional type theory. Hablitation Thesis, Ludwig Maximilian Universität, 1993.

[SUM96] H. Sipma, T. Uribe, and Z. Manna. Deductive model checking. *Proc. Inter. Conf. on Computer Aided Verification (CAV'96)*, 1996.

[SW83] D.T. Sannella and M. Wirsing. A kernal language for algebraic specification and implementation. Technical Report CSR-155-83, Dept of Computer Science, University of Edinburgh, 1983.

[Tai67] W.W. Tait. Intensional interpretation of functionals of finite type i. *J. of Symbolic Logic*, 32, 1967.

[Tai75] W. W. Tait. A realizability interpretation of the theory of species. In R. Parikh, editor, *Logic Colloquium, Lecture Notes in Mathematics, 453*, 1975.

[Tas97] A. Tasistro. *Substitution, record types and subtyping in type theory*. PhD thesis, Chalmers University of Technology, 1997.

[Tho91] S. Thompson. *Type Theory and Functional Programming*. Addison-Wesley, 1991.

[TL88] P. Taylor and Z. Luo. Theories, mathematical structures and strong sums. manuscript, December 1988.

[Tro73a] A. S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis, Lecture Notes in Mathematics, 344*, 1973.

[Tro73b] A. S. Troelstra. Notes on intuitionistic second-order arithmetic. *Lecture Notes in Mathematics, 337*, 1973.

[Tur95] D. Turner. Elementary strong functional programming. *Proceedings of the first international symposium on Functional Programming Languages in Education, LNCS 1022*, 1995.

[TvD88] A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics, an introduction*, volume I and II. North-Holland, 1988.

[vBJMP93] L. van Benthem Jutting, James McKinna, and Robert Pollack. Typechecking in pure type systems. submitted for publication, 1993.

[vD80] D. T. van Daalen. *The Language Theory of Automath*. PhD thesis, Technological University, Eindhoven, 1980.

[Wal90] L. Wallen. *Automated Proof Search in Non-classical Logics*. MIT, 1990.

[Wan92] P. Wand. Functional programming and verification with Lego. Master's thesis, Department of Computer Science, University of Edinburgh, 1992.

[Wan93] D. Wand. Elementary strong functional programming. Private communication, 1993.

[WB89] M. Wirsing and M. Broy. A modular framework for specification and implementation. *TAPSOFT'89, LNCS*, 351, 1989.

[We95] M. Wooldridge and N. Jennings (eds.). *Intelligent Agents – Theories, Architectures, and Languages, LNAI 890*. Sringer-Verlag, 1995.

[Wer97] Benjamin Werner. Sets in types, types in sets. In *Proceedings of TACS'97*, pages 530–546. Springer-Verlag, 1997.

[Wir86] M. Wirsing. Structured algebraic specifications : a kernel language. *Theoretical Computer Science*, 42 :123–249, 1986.

[WL93] D. Wang and J. R. Lee. Pictorial concepts and a concept supporting system. *Journal of Visual Languages and Computing*, 4 :177–199, 1993.

[WR25] A.N. White and B.A.W. Russell. *Principia Mathematica*. Cambridge University Press, 2nd edition, 1925.

[YBL97] E. Younger, K. Bennett, and Z. Luo. A formal transformation and refinement method for re-engineering concurrent programs. *Proc of IEEE Inter. Conf. on Software Maintenance*, 1997.

[YL97] S. Yu and Z. Luo. Implementing a model checker for Lego. *Proc. of the 4th Inter Symp. of Formal Methods Europe, FME'97 : Industrial Applications and Strengthened Foundations of Formal Methods, Graz, Austria. LNCS 1313*, 1997.

[YL98] S. Yu and Z. Luo. A verification environment based on type theory. Submitted to Journal of Formal Aspects of Computing, 1998.

[YLBB96] E. Younger, Z. Luo, K. Bennett, and T. Bull. Reverse engineering concurrent programs using formal modelling and analysis. *Proc of IEEE Inter. Conf. on Software Maintenance*, 1996.

[Yu98a] S. Yu. *Formal veirifcation of concurrent programs in type theory*. PhD thesis, University of Durham, 1998. Draft.

[Yu98b] S. Yu. A verification environment for imperative programs. Draft paper for TYPES'97, 1998.

[Zuc75] J. Zucker. Formalization of classical mathematics in AUTOMATH. *Colloque Internationaux du CNRS, 249*, 1975.

AUTEUR : Lionel Marie-Magdeleine

TITRE : Sous-typage coercitif en présence de réductions non-standards dans un système aux types dépendants

DIRECTEUR DE THESE : Sergei Soloviev

DATE ET LIEU DE SOUTENANCE : Soutenu le 11/12/2009 à l'université Paul Sabatier

---

RESUME :

La théorie des types est une discipline au croisement de la logique, des mathématiques et de l'informatique. Elle peut servir de support au développement de programme "zéro faute". L'objet de cette thèse est d'étudier l'extension d'un système aux types dépendants UTT (comprenant notamment des types inductifs) par une relation de récriture concernant un fragment du calcul, à savoir les types finis.

Nous nous assurons d'abord que les propriétés de normalisation forte, de confluence et de préservation du type sont toujours préservées malgré l'ajout de la réduction.

Ensuite nous enrichissons ce système par la notion de sous-typage coercitif vue comme un mécanisme d'abréviation et effectuons la preuve de conservativité pour le système enrichi du sous-typage par rapport au système de base.

L'intérêt d'un tel système est qu'il améliora l'efficacité des assistants à la preuve et offrira un bon cadre pour l'étude des problèmes faisant intervenir des ensembles finis (combinatoire, manipulation de graphe etc).

MOTS CLES : Théorie des types, Sous-typage coercitif, Types dépendants, Récriture.

ABSTRACT :

Type Theory lies on the crossroad of Logics, Mathematics and Computer Science. It may be used to develop the "zero-error" programs. The aim of this thesis is to study an extension of a system with dependent types called UTT (including inductive types) that is obtained by adding to the rewrite relation of UTT new rewrite rules concerning finite types.

We check that Strong Normalization, Church-Rosser property and Subject Reduction are preserved.

We consider another extension by Coercive Subtyping that is seen as an abbreviation mechanism and give a conservativity proof for the system enriched by Coercive Subtyping with respect to underlying UTT (with an without new rewrite rules).

The interest of such a system is that it will improve the efficiency of proof assistants and provides a general framework for treatment of the problems involving finite types (combinatorics, graphs etc).

DISCIPLINE : Informatique

LABORATOIRE DE RATTACHEMENT : Institut de Recherche en Informatique de Toulouse (IRIT)